

Algoritmos para Problemas de Grafos com Incertezas

Fábio Hernandes

Tese submetida à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, como parte dos requisitos exigidos para a obtenção do título de
Doutor em Engenharia Elétrica.

Orientador:

Prof. Dr. Akebo Yamakami
FEEC/UNICAMP

Co-orientadora:

Dra. Márcia Tomie Takahashi
PETROBRAS

Banca Examinadora

Profa.	Dra.	Maria José de Paula Castanho	UNICENTRO
Profa.	Dra.	Tatiane Regina Bonfim	FAC-II/Anhanguera Educacional
Profa.	Dra.	Valéria Abrão de Podestá	IMECC/UNICAMP
Prof.	Dr.	Aníbal Tavares de Azevedo	FEEC/UNICAMP - Prof. Colaborador
Prof.	Dr.	Takaaki Ohishi	FEEC/UNICAMP
Prof.	Dr.	Akebo Yamakami	FEEC/UNICAMP (presidente)

Fevereiro de 2007
FEEC - UNICAMP

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA- BAE - UNICAMP

H43a **Hernandes, Fábio**
 Algoritmos para problemas de grafos com incertezas. /
 Fábio Hernandez. – Campinas, SP: [s.n.], 2007.

Orientadores: Akebo Yamakami, Márcia Tomie
Takahashi.

Tese (doutorado) - Universidade Estadual de Campinas,
Faculdade de Engenharia Elétrica e de Computação.

1. Conjuntos Difusos. 2. Algoritmos difusos. 3. Teoria de grafos. 4. Grafos difusos. I. Yamakami, Akebo. II. Takahashi, Márcia Tomie. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Titulo em Inglês: Algorithms for fuzzy graphs problems

Palavras-chave em Inglês: Fuzzy sets, Fuzzy algorithms, Graphs theory, Fuzzy graphs

Área de concentração: Automação

Titulação: Doutor em Engenharia Elétrica

Banca examinadora: Maria José de Paula Castanho, Tatiane Regina Bonfim, Valéria
Abraão de Podestá, Aníbal Tavares de Azevedo, Takaaki Ohishi

Data da defesa: 23/02/2007

Programa de Pós-Graduação: Engenharia Elétrica

Resumo

A teoria de grafos é uma importante área da programação matemática, tendo um importante papel em áreas tais como engenharia e pesquisa operacional. Em particular, ela fornece ferramentas para tratar problemas de redes (tais como: alocação, caminho mínimo, fluxo máximo, etc.), que têm aplicações em diversas subáreas da engenharia (por exemplo: telecomunicações, transporte, manufatura, etc.). Estas aplicações podem, entretanto, possuir incertezas em seus parâmetros ou em sua estrutura. Baseado nisto, este trabalho trata de algumas importantes aplicações de problemas em grafos com incertezas em seus parâmetros ou estruturas e propõe algoritmos para encontrar suas soluções. As aplicações estudadas são: problemas de caminho mínimo, problemas de fluxo máximo, problemas de fluxo de custo mínimo e problemas de coloração de grafos. As incertezas são modeladas por meio da teoria dos conjuntos *fuzzy*, que tem sido aplicada com sucesso em problemas com incertezas e imprecisões.

Abstract

The graph theory is an important area of mathematical programming, it has an important role in fields such as engineering and operational research. In particular, it provides the tools to tackle network problems (e.g. allocation, shortest path, maximum flow, etc), which have applications in several sub areas of engineering (e.g. telecommunications, transportation, manufacturing, etc). These applications can, however, possess uncertainties in their parameters or in their structure. Based on that, this work addresses some important applications of graph problems with uncertainties in their structure or parameters and proposes algorithms to find the solution to them. The applications studied are: shortest path problems, maximum flow problems, minimum cost flow problems and graph coloring problems. The uncertainties are modeled by means of the fuzzy sets theory, which has been successfully applied to problems with uncertainties and vagueness.

DEDICO ESTA TESE AOS
MEUS PAIS, MARIA E JOSÉ.

Agradecimentos

A DEUS pelo dom da vida, por me conceder saúde e por me abençoar na realização de todos meus deveres.

À minha família, aos meus pais José e Maria, aos meus irmãos André, Ricardo e Henrique, pelo incentivo, ajuda, apoio e paciência.

Ao prof. Akebo pela orientação, disponibilidade e atenção, pela compreensão, respeito e paciência, pelo apoio, incentivo e amizade, que foram fundamentais para a realização deste trabalho.

À Márcia pela valiosa co-orientação, pela paciência, amizade e disponibilidade.

Aos professores José Luis Verdegay e Maria Teresa Lamata, da Universidade de Granada, Espanha, pela disponibilidade em aceitar-me em vossa universidade por seis meses, pela orientação, incentivo e amizade, que foram importantes para a conclusão desta tese.

Aos amigos Ricardo Ceará e Ricardo Pankadão pela valiosa ajuda durante a fase de implementação e escrita desta tese e também pela amizade e apoio.

Aos meus grandes amigos: Héder, Tatinha, Chicão, Camila, Tatiana, Maria Alice, Sandro, Gustavo, Cynthia, Adriane, Rúbia e Ricardo, pela amizade e incentivo, pelas festas e pelos momentos inesquecíveis.

Aos amigos do café: João Henrique, João Coelho, Mineirinho, Vandenberg, Vinícius, Giuliano, Renato, Valter Leite, Walter Furloni, Alessandro, Taís, Talia, Tiago, Edilson, Léo, Júlio e Gois, pelos momentos de descontração e pela amizade.

Aos professores do Departamento de Ciência da Computação da Unicentro: Josiane, Sandra, Inali, Luciane, Quináia, Bronoski, Angelita, Ana Elisa, Lucélia, Sandro e Sílvia, pelo incentivo e apoio e pela autorização do afastamento integral das atividades docentes durante estes quatro anos.

À CAPES pelo apoio financeiro.

Sumário

Resumo e Abstract	iii
Agradecimentos	vii
Lista de Figuras	xiii
Lista de Tabelas	xv
1 Introdução	1
1.1 Objetivos	2
1.2 Organização da tese	3
2 Conceitos de Teoria dos Conjuntos <i>Fuzzy</i>	5
2.1 Introdução	5
2.2 Conjuntos <i>fuzzy</i>	6
2.3 Números <i>fuzzy</i>	7
2.4 Conceitos básicos de conjuntos <i>fuzzy</i>	9
2.4.1 Definições básicas	9
2.4.2 Operações com conjuntos <i>fuzzy</i>	10
2.5 Relações <i>fuzzy</i> sobre conjuntos <i>fuzzy</i>	11
2.6 Teoria de possibilidade	12
2.7 Relações de ordens	13
2.8 Considerações finais	17
3 Problemas de Grafos <i>Fuzzy</i>	19
3.1 Introdução	19
3.2 Problemas com estrutura do grafo <i>crisp</i> e parâmetros <i>fuzzy</i>	20
3.2.1 Abordagem de Bellman e Zadeh, 1970	20
3.2.2 Abordagem de Okada, 2004	21
3.3 Problemas com estrutura de grafos <i>fuzzy</i>	23

3.3.1	O trabalho de Delgado et. al, 1990	24
3.4	Considerações finais	26
4	O Problema de Caminho Mínimo com Incertezas	27
4.1	Introdução	27
4.2	Revisão bibliográfica	28
4.3	Formulação matemática	30
4.4	Algoritmo proposto	31
4.4.1	Relação de ordem de Okada e Soper	31
4.4.2	Apresentação do algoritmo	31
4.4.3	Estudo da complexidade e da convergência	34
4.4.3.1	Comentários gerais	34
4.4.3.2	Estudo da otimalidade	34
4.4.3.3	Complexidade e convergência	34
4.4.4	Resultados computacionais	35
4.4.5	Conclusões da Seção 4.4	37
4.5	Problema de caminho mínimo com incertezas e restrições de tempo	38
4.5.1	Restrições de tempo	38
4.5.2	Algoritmo com restrição de tempo	38
4.5.3	Resultados computacionais	41
4.5.4	Conclusões da Seção 4.5	43
4.6	Uso da similaridade entre caminhos mínimos com incertezas	43
4.6.1	Heurística de Chuang e Kung	43
4.6.1.1	Procedimento para encontrar o CCMF (L_{min})	44
4.6.1.2	Procedimento heurístico para detectar o CCMF (L_{min})	46
4.6.1.3	Algoritmo proposto por Chuang e Kung	48
4.6.2	Adaptação do algoritmo proposto	48
4.6.3	Resultados computacionais	49
4.6.4	Conclusões da Seção 4.6	52
4.7	Algoritmo genérico para o problema de caminho mínimo com incertezas	52
4.7.1	Relação de ordem	53
4.7.2	Algoritmo com relação de ordem genérica	53
4.7.3	Resultados computacionais	55
4.7.4	Conclusões da Seção 4.7	58
4.8	Considerações finais	58

5	O Problema de Fluxo Máximo com Incertezas	59
5.1	Introdução	59
5.2	O problema de fluxo máximo	61
5.2.1	O problema de fluxo máximo clássico	62
5.2.2	O problema de fluxo máximo <i>fuzzy</i>	63
5.3	Algoritmos para o problema de fluxo máximo com incertezas	65
5.3.1	Algoritmo proposto por Chanas e Kolodziejczyk (1984)	65
5.3.2	Algoritmo Proposto	66
5.4	Resultados computacionais	69
5.5	Considerações finais	71
6	O Problema de Fluxo de Custo Mínimo com Incertezas	73
6.1	Introdução	73
6.2	Formulação do problema de fluxo de custo mínimo (PFCM)	75
6.2.1	Formulação do PFCM clássico	75
6.2.2	Formulação do PFCM <i>fuzzy</i>	75
6.3	Algoritmos propostos	76
6.3.1	Adaptação do método do Big-M	76
6.3.2	Algoritmos propostos para problemas com incertezas	77
6.3.2.1	Algoritmo para o PFCM-UOUD	78
6.3.2.2	Algoritmo para o PFCM-UOMD ou PFCM-MOUD	81
6.3.2.3	Algoritmo para o PFCM-MOMD	82
6.4	Resultados computacionais	83
6.4.1	Resultados do algoritmo adaptado do Big-M	83
6.4.2	Resultados do PFCM-UOUD	85
6.4.3	Resultados do PFCM-UOMD	89
6.4.4	Resultados do PFCM-MOUD	90
6.4.5	Resultados do PFCM-MOMD	91
6.5	Considerações finais	91
7	Coloração em Grafos <i>Fuzzy</i>	93
7.1	Introdução	93
7.2	Conceitos preliminares	94
7.3	O problema da coloração <i>fuzzy</i>	95
7.3.1	O problema dos semáforos	95
7.3.2	A função coloração de um grafo <i>fuzzy</i>	98
7.4	O problema da (d,f) -coloração	100

7.4.1	O problema da distribuição de exames	100
7.4.2	A coloração (d,f) -estendida de um grafo <i>fuzzy</i>	102
7.5	O problema da (d,f,g) -coloração	105
7.5.1	O problema da alocação de horários de disciplinas	106
7.5.2	A coloração (d,f,g) -estendida de um grafo <i>fuzzy</i>	107
7.6	Algoritmos propostos	108
7.6.1	Algoritmo proposto por Muñoz et al para a (d,f) -coloração	108
7.6.2	Algoritmo proposto para o problema da (d,f,g) -coloração	110
7.7	Resultados computacionais	114
7.8	Considerações finais	115
8	Conclusões e Trabalhos Futuros	117
8.1	Conclusões	117
8.2	Trabalhos futuros	119
8.3	Trabalhos publicados	120
	Bibliografia	122
A	Conceitos de Grafos Clássicos	129
B	Problemas de Grafos Clássicos	135
B.1	Problema de caminho mínimo	135
B.2	Problemas de fluxo máximo	136
B.3	Problemas de fluxo de custo mínimo (PFCM)	137
B.3.1	Algoritmo do Big-M	137
B.3.2	Algoritmo Fase II	141
B.4	Problema de coloração de grafos	141

Lista de Figuras

2.1	Exemplo de uma função característica clássica	7
2.2	Exemplo de uma função de pertinência	7
2.3	Exemplo de uma função de pertinência triangular	8
2.4	Exemplo de uma função de pertinência trapezoidal	9
2.5	Exemplo de α -cortes	10
2.6	Operações com conjuntos <i>fuzzy</i>	11
2.7	Exemplo de dominância parcial	16
3.1	Exemplo de grafo com parâmetros <i>fuzzy</i>	20
3.2	Exemplo de um gráfico do modelo de Bellman e Zadeh	21
3.3	Exemplo - caminho mínimo	22
3.4	Caminhos do grafo da Figura 3.3	22
3.5	Exemplo de um grafo com estrutura <i>fuzzy</i>	24
4.1	Rede óptica italiana	35
4.2	Rede com custos negativos	37
4.3	Rede européia - COST239	41
4.4	x à direita de m	45
4.5	Exemplo L'_k	46
4.6	Rede de Chuang e Kung	50
4.7	Rede exemplo	56
4.8	Rede negativa	57
5.1	Exemplo de capacidade <i>fuzzy</i>	63
5.2	Exemplo de uma função que representa $\mu_S(v)$	64
5.3	Exemplo de função de satisfação	65
5.4	Rede CK(84)	69
5.5	Resultados - algoritmo proposto	71
5.6	Resultados - Chanas e Kolokziejczyk	72

6.1	Ordenação dos caminhos p_1, p_2, p_3	79
6.2	Rede exemplo	79
6.3	Capacidade do arco (1,2)	80
6.4	Rede PFCM-Clássico	84
6.5	Rede PFCM-uma entrada e uma saída	85
6.6	Rede PFCM-uma entrada e múltiplas saídas	89
6.7	Rede PFCM-múltiplas entradas e uma saída	90
6.8	Rede PFCM-múltiplas entradas e múltiplas saídas	91
7.1	Exemplo de uma 3-coloração	95
7.2	Exemplo de um cruzamento entre ruas	96
7.3	Representação da Figura 7.2 em forma de grafo	96
7.4	Representação do grafo de incompatibilidade do Exemplo 7.1	98
7.5	Representação do grafo $\tilde{G}$	101
7.6	Representação do grafo $\tilde{G}$ do Exemplo 7.4	104
7.7	Grafo $\tilde{G}$ do Exemplo 7.5	107
7.8	Grafo de incompatibilidades do Exemplo 7.7	114
A.1	Exemplo de um arco (i,j)	129
A.2	Exemplo de uma aresta (i,j)	130
A.3	Exemplo de um grafo orientado	130
A.4	Exemplo de um grafo não-orientado	130
A.5	Exemplo de um laço	130
A.6	Exemplo de um caminho	131
A.7	Exemplo de um circuito	131
A.8	Exemplo de um ciclo	132
A.9	Exemplo de nós capacitados	133
A.10	Exemplo de um conjunto de corte $[X, \bar{X}]$	133
A.11	Exemplo de um grafo desconexo	133

Lista de Tabelas

4.1	Denominação dos nós da rede italiana	36
4.2	Tabela dos arcos e custos da rede italiana	36
4.3	Denominação dos nós da rede (COST239) e restrições de tempo	41
4.4	Tabela dos dados da rede COST239	42
4.5	Resultados da rede COST239	42
4.6	Tabela dos arcos e custos da rede italiana (Figura 4.1)	51
4.7	Custos dos arcos do Exemplo 4.6	55
4.8	Resultados da rede com custos positivos (Exemplo 4.6).	56
4.9	Resultados da rede negativa - Exemplo 4.7	57
5.1	Resultados do algoritmo proposto para $\bar{\alpha} = 0$ e $H = 10$	70
5.2	Resultados do algoritmo proposto para $\bar{\alpha} = 0.5$ e $H = 2$	70
6.1	Capacidades dos arcos da Figura 6.2	79
6.2	Custos e capacidades da Figura 6.4	84
6.3	Valores dos α_k	84
6.4	Valores dos custos defuzzificados	84
6.5	Fluxo final	85
6.6	Custos e capacidades da Figura 6.5	85
6.7	Custos e capacidades da Figura 6.6	90
6.8	Custos e capacidades da Figura 6.7	90
7.1	α , A_α , X_α e $C_\alpha^{X_\alpha}$ do Exemplo 7.1	99
7.2	Resultados da Figura 7.5	102
7.3	Função escala f do Exemplo 7.3	103
7.4	Medida de dissimilaridade d do Exemplo 7.4	104
7.5	Função escala f do Exemplo 7.4	104
7.6	Resultados do Exemplo 7.2	105
7.7	Resultados da Figura 7.7	107
7.8	Resultado do Exemplo 7.5.	108

7.9	Resultado do Exemplo 7.6 usando (d, f) -coloração	114
-----	---	-----

Capítulo 1

Introdução

A teoria dos grafos clássica é uma área da Programação Matemática bastante estudada, visto que existe uma grande quantidade de problemas reais que podem ser representados na forma de grafos (caminho mínimo, fluxo máximo, fluxo de custo mínimo, coloração de grafos, dentre outros) com aplicações nas mais diversas áreas da engenharia (energia, telecomunicações, transportes, manufatura, etc). Nos livros de Gondran e Minoux (1984), Goldbarg e Luna (2000), Ahuja, Magnanti e Orlin (1993) e Bazarra, Jarvis e Sherali (1990) são encontrados estes problemas, bem como os vários e eficientes algoritmos existentes para resolvê-los.

Normalmente, problemas reais têm associado uma série de parâmetros que não são naturalmente precisos, tais como: custo, capacidade e demanda. Além disso, podem existir casos em que nem a estrutura do grafo em si é precisa. Isto faz com que uma modelagem sob o ponto de vista da teoria dos conjuntos *fuzzy* seja interessante.

A introdução da teoria dos conjuntos *fuzzy* na programação matemática foi proposta por Zadeh (1965). Porém, foi somente na última década que houve uma intensificação nos estudos desta área, onde foram abordadas extensões das teorias que envolvem programação linear, não-linear e inteira mista para um ambiente impreciso. Também foram desenvolvidas novas áreas como a da programação possibilística. No trabalho de Inuiguchi e Ramik (2000) são descritos e classificados alguns problemas de programação matemática *fuzzy*.

Como já comentado, a teoria dos grafos clássica possui um vasto número de trabalhos, porém, o estudo em grafos com incertezas, mais especificamente em *grafos fuzzy*, ainda se encontra na fase inicial, tanto em sua parte teórica quanto em suas aplicações. Esta teoria

é extensa, pois existe uma grande quantidade de problemas que podem ser representados na forma de grafos *fuzzy*. Existem diferentes formas de abordar as incertezas do problema, pois estas podem estar tanto na estrutura (nós e/ou arcos) quanto nos parâmetros (custos, capacidades e demandas). Quanto às suas soluções, há diferentes formatos para tais, sendo que alguns apresentam soluções únicas enquanto outros apresentam um conjunto. Vários problemas apresentam complexidades altas, do tipo NP-completos, o que torna difícil suas resoluções.

Considerando as muitas aplicações de grafos *fuzzy* na engenharia e o reduzido número de trabalhos na área, nesta tese são estudados os principais problemas de grafos *fuzzy* (Capítulo 3), bem como algumas extensões de aplicações da teoria de grafos clássica para a teoria de conjuntos *fuzzy*, propondo novos algoritmos.

Vale ressaltar a existência de outra área da teoria dos conjuntos *fuzzy* também chamada *grafos fuzzy*, que não será tratada neste trabalho, onde são abordados trabalhos da área de metamodelagem. Este conceito foi introduzido por Zadeh (1997), sendo definido como uma disjunção de produtos cartesianos de conjuntos *fuzzy*. Esta definição pode ser encontrada no trabalho de Tan, Yu e Wang (1998).

A seguir são apresentados os objetivos da tese e a organização da mesma.

1.1 Objetivos

Takahashi (2004) desenvolveu sua tese sobre este assunto. Em seu trabalho, a autora fez um estudo sobre a teoria dos grafos *fuzzy* proposta por Rosenfeld (1975) e a extensão de aplicações da teoria de grafos clássica para a teoria dos conjuntos *fuzzy*.

Neste trabalho são abordados novos conceitos de grafos *fuzzy*. Através de levantamento bibliográfico, são propostos algoritmos para problemas conhecidos em teoria dos grafos, mas pouco estudados quando existe incerteza associada ao problema. Também são estudados problemas conhecidos em grafos *fuzzy* e são propostos algoritmos mais abrangentes. Os problemas abordados são:

- caminho mínimo em grafos com estrutura *crisp* e parâmetros *fuzzy*;

- fluxos em redes com parâmetros *fuzzy*: o problema de fluxo de custo mínimo com capacidades e custos *fuzzy* e o problema de fluxo máximo em grafos com capacidades *fuzzy*; e
- coloração de grafos *fuzzy*.

1.2 Organização da tese

Esta tese está dividida em oito capítulos, onde são estudados os principais problemas da literatura que abordam incertezas nos grafos, tanto nos parâmetros quanto nas estruturas.

Os Capítulos 2 e 3 tratam, respectivamente, de conceitos básicos da teoria dos conjuntos *fuzzy* e de problemas que envolvem grafos *fuzzy*.

O Capítulo 4 trata do problema de caminho mínimo em grafos com estrutura *crisp* e parâmetros *fuzzy*. Nesse capítulo é feita uma minuciosa revisão bibliográfica dos trabalhos da literatura, além da proposta de alguns algoritmos que contornam dificuldades dos trabalhos existentes. São propostos algoritmos para problemas que possuem um parâmetro incerto (por exemplo: custo) e também para problemas que envolvem dois parâmetros incertos (por exemplo: custos e tempo).

O problema de fluxo máximo em grafos com estrutura *crisp* e parâmetros *fuzzy* é tratado no Capítulo 5, onde são abordados os principais trabalhos da literatura e também é proposto um algoritmo para este problema.

No Capítulo 6 é tratado o problema de fluxo de custo mínimo em grafos com estrutura *crisp* e parâmetros incertos. É feito um levantamento bibliográfico do tema e também são propostos alguns algoritmos para tal, sendo que, um destes algoritmos transforma o problema incerto em um clássico, resolvendo-o, assim, através de algoritmos da literatura. Já os demais algoritmos trabalham com problemas com incertezas.

O problema de coloração em grafos com estrutura incerta é abordado no Capítulo 7, onde é proposta uma nova definição de coloração *fuzzy*, bem como um algoritmo para resolvê-lo.

No Capítulo 8 são apresentadas as conclusões e propostas de trabalhos futuros.

Finalmente, os Apêndices A e B trazem os principais conceitos e problemas de grafos clássicos.

Capítulo 2

Conceitos de Teoria dos Conjuntos *Fuzzy*

2.1 Introdução

A teoria dos conjuntos *fuzzy* foi introduzida, em 1965, pelo matemático Lotfi A. Zadeh (Zadeh 1965, Zadeh 1968, Zadeh 1978), com a principal intenção de dar um tratamento matemático a certos termos lingüísticos subjetivos, como “aproximadamente”, “em torno de”, dentre outros. Esse seria um primeiro passo para se representar e armazenar informações incertas, tornando possível o cálculo com estas informações, a exemplo do que faz o ser humano. Devido a essa possibilidade de manipulação com informações incertas, a teoria dos conjuntos *fuzzy* tem se tornado uma das áreas emergentes na tecnologia contemporânea. Nas engenharias, os chamados controladores *fuzzy* têm sido largamente utilizados em eletrodomésticos, com o objetivo de imitar o homem na execução de algumas tarefas. Como nesta tese são abordados os problemas em grafos com incertezas, estas serão tratadas utilizando a teoria dos conjuntos *fuzzy*.

Muitas vezes o conceito de imprecisão associado às informações processadas pelos conjuntos *fuzzy* é confundido com a incerteza relacionada à teoria da probabilidade. A principal diferença entre a imprecisão nos conjuntos nebulosos e a incerteza probabilística é que a primeira trata de possibilidade (plausibilidade) determinística, ao passo que a segunda lida com a verossimilhança de eventos determinísticos ou estocásticos (Berthold e Hand 1999). A incerteza que está por trás do conceito de probabilidade geralmente se relaciona com a

ocorrência de um fenômeno simbolizado pelo conceito de aleatoriedade, como, por exemplo, quando se tenta determinar "probabilidade de alguma pessoa escolhida aleatoriamente dentro de uma população ser jovem", ou "a probabilidade do próximo carro que cruza a avenida estar em velocidade alta". Por outro lado, a imprecisão nos conjuntos *fuzzy* está relacionada à ambigüidade encontrada na definição de conceitos lingüísticos como "pessoa jovem" ou "velocidade alta" (maiores detalhes: (Delgado 2002, Pedrycz e Gomide 1998)).

Como nesta tese as incertezas são tratadas utilizando a teoria dos conjuntos *fuzzy*, neste capítulo são introduzidos os conceitos básicos desta teoria, bem como a definição da teoria de possibilidade. As obras de Dubois e Prade (1980) e Pedrycz e Gomide (1998) podem ajudar a aprofundar o estudo desta teoria.

Este capítulo está organizado da seguinte forma: a Seção 2 apresenta a definição de conjunto *fuzzy*. Na Seção 3 são definidos os números *fuzzy* que são utilizados na tese e também algumas operações entre eles. Na Seção 4 estão alguns conceitos sobre conjuntos *fuzzy*. Na Seção 5 é definida relação *fuzzy*. A definição da teoria de possibilidade está na Seção 6. Na Seção 7 são abordadas as relações de ordens utilizadas na tese. A Seção 8 apresenta as considerações finais.

2.2 Conjuntos *fuzzy*

Conjunto é uma tentativa de organizar, resumir e generalizar conhecimento sobre objetos. Neste sentido trabalha-se 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 $H = \{x \in \mathbb{R} | x \geq 1,80\}$. Alguém com 1,79m não pode ser considerado alto, ao passo que alguém com 1,81m é seguramente alto (Figura 2.1).

Um subconjunto *fuzzy* A em X é definido por uma *função de pertinência* μ_A que associa a cada ponto de X um número real no intervalo $[0,1]$, com o valor de μ_A em x representando

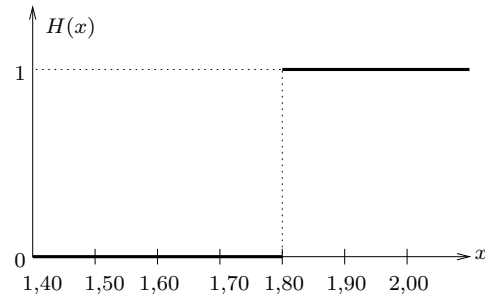


Figura 2.1: Exemplo de uma função característica clássica

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 (Figura 2.2).

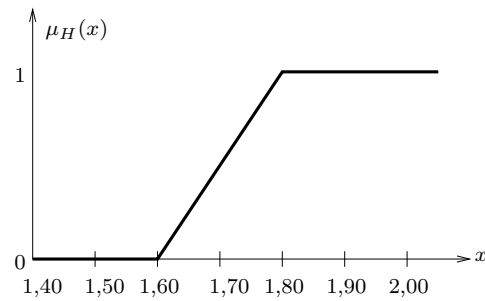


Figura 2.2: Exemplo de uma função de pertinência

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 (Figura 2.1), será chamado de conjunto clássico (ordinário ou *crisp*) ou simplesmente conjunto.

2.3 Números *fuzzy*

Definição 1 Um número fuzzy, denotado por $\tilde{a} = (m, \alpha, \beta)$, é um número triangular fuzzy se sua função de pertinência, $\mu_{\tilde{a}}(x)$, for definida da seguinte forma (Figura 2.3):

$$\mu_{\tilde{a}}(x) = \begin{cases} 0, & \text{se } x \leq m - \alpha \\ \frac{x - (m - \alpha)}{\alpha}, & \text{se } m - \alpha < x < m \\ 1, & \text{se } x = m \\ \frac{(m + \beta) - x}{\beta}, & \text{se } m < x < m + \beta \\ 0, & \text{se } x \geq m + \beta \end{cases} \quad (2.2)$$

tal que:

m : valor modal;

α : espalhamento à esquerda; e

β : espalhamento à direita.

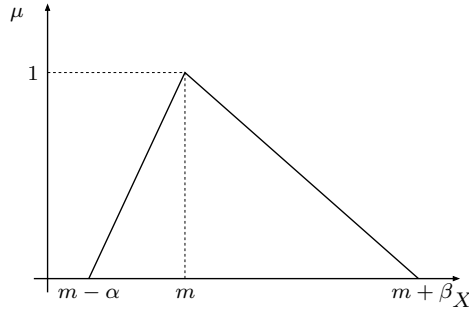


Figura 2.3: Exemplo de uma função de pertinência triangular

Ressalta-se que os valores $(m - \alpha)$ e $(m + \beta)$ são chamados de limitantes inferior e superior, respectivamente.

Definição 2 Valor modal é o valor $x \in [m - \alpha, m + \beta]$ para o qual a função de pertinência tem valor máximo.

Definição 3 Um número fuzzy trapezoidal (Figura 2.4), denotado por $\tilde{a} = (m_1, m_2, \alpha, \beta)$, possui sua função de pertinência, $\mu_{\tilde{a}}(x)$, definida da seguinte forma:

$$\mu_{\tilde{a}}(x) = \begin{cases} 0, & \text{se } x \leq m_1 - \alpha \\ \frac{x - (m_1 - \alpha)}{\alpha}, & \text{se } m_1 - \alpha < x < m_1 \\ 1, & \text{se } m_1 \leq x \leq m_2 \\ \frac{(m_2 + \beta) - x}{\beta}, & \text{se } m_2 < x < m_2 + \beta \\ 0, & \text{se } x \geq m_2 + \beta \end{cases} \quad (2.3)$$

tal que:

m_1 : extremo inferior do valor modal;

m_2 : extremo superior do valor modal;

α : espalhamento à esquerda; e

β : espalhamento à direita.

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

$$\tilde{a} + \tilde{b} = (m_1, \alpha_1, \beta_1) \oplus (m_2, \alpha_2, \beta_2) = (m_1 + m_2, \alpha_1 + \alpha_2, \beta_1 + \beta_2) \quad (2.4)$$

Quanto à soma de números trapezoidais, esta é similar.

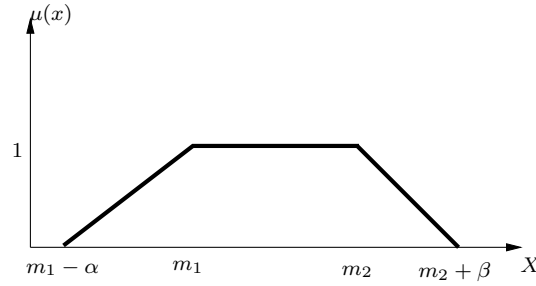


Figura 2.4: Exemplo de uma função de pertinência trapezoidal

2.4 Conceitos básicos de conjuntos *fuzzy*

Nesta seção estão alguns conceitos básicos dos conjuntos *fuzzy* que são necessários no decorrer do texto.

2.4.1 Definições básicas

Definição 5 Um conjunto fuzzy S é normal se sua função de pertinência possuir pelo menos um valor tal que $\mu_S(x) = 1$.

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

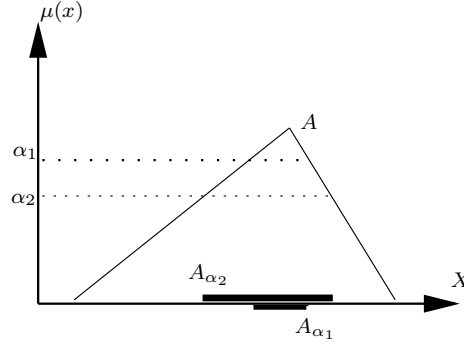
$$\text{supp}(S) = \{x \in X | \mu_S(x) > 0\} \quad (2.5)$$

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

Definição 7 O núcleo de um conjunto fuzzy S é o conjunto de todos os elementos cujo valor de pertinência é igual a 1

$$\text{nucleo}(S) = \{x \in X | \mu_S(x) = 1\} \quad (2.6)$$

Definição 8 O α -corte de um subconjunto fuzzy é um conjunto $[\tilde{a}]_\alpha = \{x | \mu_{\tilde{a}}(x) \geq \alpha\}$, tal que $\alpha \in [0, 1]$ (Figura 2.5).

Figura 2.5: Exemplo de α -cortes

Definição 9 Um conjunto fuzzy S é convexo se sua função de pertinência é tal que

$$\mu_S[\lambda x_1 + (1 - \lambda)x_2] \geq \min[\mu_S(x_1), \mu_S(x_2)] \quad (2.7)$$

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

2.4.2 Operações com conjuntos fuzzy

A união, a intersecção e o complemento são operações essenciais realizadas em conjuntos clássicos. Com base nisto, Zadeh (1965) definiu, a partir da função de pertinência, estas operações para conjuntos fuzzy.

Definição 10 A união de dois conjuntos fuzzy A e B (Figura 2.6) é dada da seguinte forma:

$$\mu_{(A \cup B)}(x) = \max[\mu_A(x), \mu_B(x)] = \mu_A(x) \vee \mu_B(x) \quad (2.8)$$

Definição 11 A intersecção de dois conjuntos fuzzy A e B (Figura 2.6) é dada da seguinte forma:

$$\mu_{(A \cap B)}(x) = \min[\mu_A(x), \mu_B(x)] = \mu_A(x) \wedge \mu_B(x) \quad (2.9)$$

Definição 12 O complemento de um conjunto fuzzy A (Figura 2.6) é dado da seguinte forma:

$$\mu_{(\bar{A})}(x) = 1 - \mu_A(x) \quad (2.10)$$

A intersecção e a união podem ser identificadas pela conjunção (E) e pela disjunção (OU), respectivamente. Assim, estas operações podem ser representadas pelos operadores $\wedge$ e $\vee$ (Pedrycz e Gomide 1998).

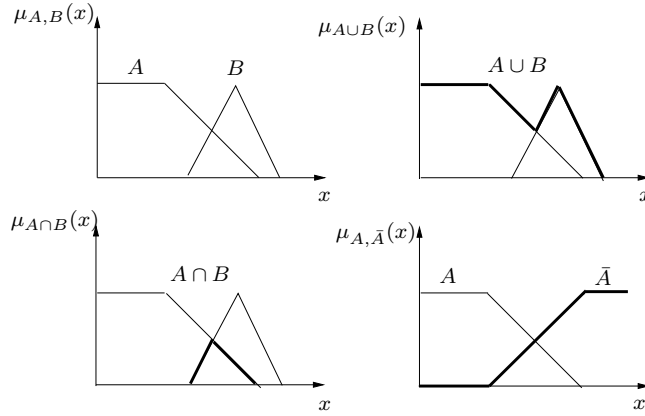


Figura 2.6: Operações com conjuntos fuzzy

2.5 Relações fuzzy sobre conjuntos fuzzy

Seja X um conjunto. Um subconjunto fuzzy S 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

$$R(x, y) = 1 \text{ implica que } T(x) = T(y) = 1, \quad \forall x, y \in X$$

Pode-se 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 S é um subconjunto *fuzzy* de X e σ sua função de pertinência, e sendo R uma relação *fuzzy* sobre X , diz-se que R é uma relação *fuzzy* sobre S se

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

Em outras palavras, para R ser uma relação *fuzzy* sobre S , é necessário que o grau de pertinência de um par de elementos nunca exceda o grau de pertinência dos próprios elementos. Se for considerado os elementos como sendo os nós de um grafo e os pares como os arcos, isto equivale ao requisito de que a pertinência de um arco nunca exceda as pertinências de seus nós.

2.6 Teoria de possibilidade

Seja um grafo $G = (V, A)$ (definições de grafos encontram nos Apêndices A e B) com custo $\tilde{c} \in \mathfrak{R}^n$, $\tilde{c} = \{\tilde{c}_{ij}\}$, associados às suas arestas. Sejam dois sub-grafos *fuzzy* T^1 e T^2 , $T^1 \neq T^2$. Pode-se dizer que T^1 tem um grau de possibilidade de ser menor do que T^2 dado por (Okada 2004):

$$\tilde{w} = Poss\left(\sum_{(i,j) \in T^1} \tilde{c}_{ij} \leq \sum_{(i,j) \in T^2} \tilde{c}_{ij}\right) = \sup_{u \leq v} \min\{\mu_{T^1}(u), \mu_{T^2}(v)\} \quad (2.11)$$

Esta equação também é estudada em Dubois e Prade (1983).

Para encontrar uma solução *fuzzy* utilizando a teoria de possibilidade é necessário 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 2004). O grau de possibilidade de T ser a solução ótima é dado pela fórmula:

$$D_T = \min_{T^k \in \mathcal{T}} \left\{ Poss\left(\sum_{(i,j) \in T^k} \tilde{c}_{ij} \leq \sum_{(i,j) \in T} \tilde{c}_{ij}\right) \right\} \quad (2.12)$$

tal que: $\mathcal{T}$ é o conjunto de todas as soluções.

e o grau de possibilidade do arco (i, j) pertencer à solução ótima é dado pela equação:

$$D_{ij} = \max_{T^k | (i,j) \in T^k} \{D_{T^k}\} \quad (2.13)$$

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 2.1 *Dados dois números fuzzy, normalizados, $\tilde{a}$ e $\tilde{b}$, tem-se 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: Consultar Takahashi (2004) ■

Teorema 2.1 *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 T^* , associada à solução crisp T^0 , com custo igual a $\tilde{w}^* = \sum_{ij \in T^*} \tilde{c}_{ij}$. Se T^* for a solução ótima do problema, então $Poss(\sum_{ij \in T^*} \tilde{c}_{ij} \leq \sum_{ij \in T^k} \tilde{c}_{ij}) = 1$, sendo T^k qualquer solução em G .*

Prova: Consultar Takahashi (2004) ■

Proposição 2.2 *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: Consultar Takahashi (2004) ■

2.7 Relações de ordens

No Capítulo 4 será abordado o problema de caminho mínimo com incertezas, sendo necessária a comparação entre números *fuzzy*, logo, nesta seção, são abordadas as relações de ordens (comparações entre números *fuzzy*) utilizadas.

a) Primeiro índice de Yager (Yager 1978, Yager 1980, Yager 1981)

Este índice é chamado de centróide, tendo a seguinte fórmula:

$$f(\tilde{a}) = \frac{\int_0^1 \alpha \tilde{a}_\alpha d\alpha}{\int_0^1 \tilde{a}_\alpha d\alpha} \quad (2.14)$$

b) Índice de Liou e Wang (1992)

Em (Liou e Wang 1992) foi proposto um método de defuzzificação que trabalha com as áreas dos espalhamentos dos números *fuzzy* triangulares (espalhamento à esquerda e à direita). O índice proposto é o seguinte:

Definição 13 *Dado um número fuzzy $\tilde{a} = (m, \alpha, \beta)$, o índice de Liou e Wang é definido como:*

$$LW^\lambda(\tilde{a}) = \lambda S_D(\tilde{a}) + (1 - \lambda) S_E(\tilde{a}) \quad (2.15)$$

tal que:

S_D : área do espalhamento à direita;

S_E : área do espalhamento à esquerda; e

$\lambda \in [0, 1]$ é o grau que reflete o otimismo e o pessimismo do decisor.

c) Índice de García e Lamata (2005)

García e Lamata propuseram um índice que além de trabalhar com o grau de otimismo do usuário, semelhante ao índice de Liou e Wang, também inclui um grau de modalidade do usuário.

Definição 14 *Seja um número fuzzy $\tilde{a} = (m, \alpha, \beta)$, o índice de García e Lamata é definido como:*

$$I(\tilde{a}) = (1 - \delta)[\lambda S_D(\tilde{a}) + (1 - \lambda) S_E(\tilde{a})] + \delta m \quad (2.16)$$

tal que: $S_D(\tilde{a})$ e $S_E(\tilde{a})$ estão definidos em Liou e Wang, $\delta, \lambda \in [0, 1]$ são, respectivamente, os graus de modalidade e otimismo do decisor.

d) Relação de ordem de Okada e Soper (Okada e Soper 2000)

Como já comentado, o algoritmo de Okada e Soper tem como finalidade encontrar os caminhos não-dominados, isto é, Pareto-ótimos de uma rede. Nesse algoritmo foram utilizados números *fuzzy* triangulares e foi determinado o seguinte critério de dominância *fuzzy*:

Definição 15 *Sejam $\tilde{a} = (m_1, \alpha_1, \beta_1)$ e $\tilde{b} = (m_2, \alpha_2, \beta_2)$ dois números fuzzy triangulares, então $\tilde{a} \prec \tilde{b}$ ($\tilde{a}$ domina $\tilde{b}$) se e somente se $m_1 \leq m_2$, $(m_1 - \alpha_1) \leq (m_2 - \alpha_2)$, $(m_1 + \beta_1) \leq (m_2 + \beta_2)$ e $\tilde{a} \neq \tilde{b}$.*

Juntamente com a definição de Okada e Soper, foi implementada a definição de dominância parcial (Okada e Soper 2000, Kuchta 2002).

Definição 16 *Sejam um número triangular fuzzy $\tilde{a} = (m, \alpha, \beta)$ e um escalar $\varepsilon \in [0, 1]$. O conjunto $\{x \in \mathfrak{R} \mid \mu_{\tilde{a}}(x) \geq \varepsilon\}$ é um intervalo fechado, denotado por $[(m - \alpha)_\varepsilon, (m + \beta)_\varepsilon]$, e chamado de ε -corte de um número fuzzy triangular $\tilde{a}$ para $\varepsilon > 0$. O zero-corte é definido como $[(m - \alpha), (m + \beta)]$.*

Definição 17 *Sejam $\tilde{a} = (m_1, \alpha_1, \beta_1)$ e $\tilde{b} = (m_2, \alpha_2, \beta_2)$ dois números triangulares fuzzy e $\varepsilon \in [0, 1]$, então $\tilde{a}$ domina $\tilde{b}$ com grau ε , denotado por $\tilde{a} \prec_\varepsilon \tilde{b}$, se e somente se $m_1 \leq m_2$, $(m_1 - \alpha_1)_\varepsilon \leq (m_2 - \alpha_2)_\varepsilon$, $(m_1 + \beta_1)_\varepsilon \leq (m_2 + \beta_2)_\varepsilon$ e $\tilde{a} \neq \tilde{b}$.*

A Figura 2.7 exemplifica a definição anterior. Para valores maiores ou iguais a ε , $\tilde{b}$ domina $\tilde{a}$ e para valores menores que ε nenhum dos dois números domina o outro.

e) Índice de aceitabilidade de Nayeem e Pal (2005)

O índice de aceitabilidade proposto por Nayeem e Pal é definido para números *fuzzy* triangulares, sendo uma extensão do índice proposto por Sengupta e Pal (2000) para números intervalares. Tal índice é definido da seguinte forma:

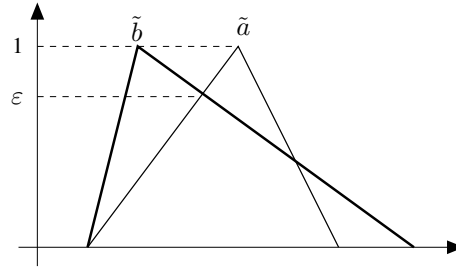


Figura 2.7: Exemplo de dominância parcial

Definição 18 Dados dois números triangulares fuzzy $\tilde{a} = (m_1, \alpha_1, \beta_1)$ e $\tilde{b} = (m_2, \alpha_2, \beta_2)$, $\tilde{a}$ domina $\tilde{b}$ se e somente se:

$$A(\tilde{a} \prec \tilde{b}) = \frac{m_2 - m_1}{\beta_1 + \alpha_2} \geq 1 \quad (2.17)$$

Se $0 < A(\tilde{a} \prec \tilde{b}) < 1$ então $\tilde{a}$ domina $\tilde{b}$ parcialmente.

f) Possibilidade de Dubois e Prade (1983)

O índice de possibilidade proposto por Dubois e Prade é definido da seguinte forma:

Definição 19 Dados dois números triangulares fuzzy $\tilde{a} = (m_1, \alpha_1, \beta_1)$ e $\tilde{b} = (m_2, \alpha_2, \beta_2)$, $\tilde{a}$ domina $\tilde{b}$ se e somente se:

$$Poss(\tilde{a} < \tilde{b}) > Poss(\tilde{b} < \tilde{a}) \quad (2.18)$$

Vale ressaltar que a medida de possibilidade de Dubois e Prade está definida na seção 2.6.

Conclusões sobre as relações de ordens

Analisando as relações citadas anteriormente, conclui-se que:

- Se o usuário desejar um índice de defuzzificação que utilize o valor médio (centro de massas), este deverá escolher o primeiro índice de Yager (Yager 1978, Yager 1980, Yager 1981).
- Caso esteja interessado em trabalhar com índices de defuzzificação que utilizam os espalhamentos à esquerda ou à direita, abordando otimismo ou pessimismo, é interessante utilizar o índice de Liou e Wang (1992).

- Se desejar índices que utilizam a modalidade, abordando pessimismo ou otimismo, recomenda-se o índice de García e Lamata (2005).
- Caso o interesse seja em um conjunto solução de caminhos e não em somente um único caminho, é interessante o uso da relação de ordem de Okada e Soper (2000).
- Se o interesse estiver no valor modal, recomenda-se o índice de Nayeem e Pal (2005).
- Caso deseje trabalhar a medida de possibilidade, usa-se a Dubois e Prade (1983).

Com base nos comentários anteriores, o usuário poderá optar por uma das relações comentadas, ou propor uma outra.

2.8 Considerações finais

Neste capítulo foram apresentados os conceitos básicos de teoria dos conjuntos *fuzzy* utilizados na tese. Maiores detalhes, consultar: Pedrycz e Gomide (1998) e Takahashi (2004).

Capítulo 3

Problemas de Grafos *Fuzzy*

3.1 Introdução

Em um problema de grafos *fuzzy*, a característica *fuzzy* do problema pode ser encontrada em dois níveis: na estrutura do grafo (nós e arcos) e nos parâmetros associados (custos, capacidades, etc). No problema de fluxo em uma rede, pode-se associar incerteza ao valor das demandas e dos fluxos. Também existem diferentes formatos na literatura para abordar o problema, dependendo do objetivo do decisor.

Na área de programação matemática, o trabalho de Rosenfeld (1975) figura entre os primeiros exemplos da utilização da teoria de grafos *fuzzy*. Um grafo, no contexto da teoria de conjuntos *fuzzy*, pode apresentar incertezas em diferentes níveis. Existem, entretanto, dois tipos de problemas que são estudados com mais frequência:

- problemas com estrutura bem definida, modelada por meio de um grafo clássico (*crisp*), e incerteza em um ou mais parâmetros. Estas incertezas podem estar associadas a parâmetros tais como: custo, capacidade e tempo. Esses parâmetros são tratados como imprecisos por natureza e modelados como números *fuzzy*; e
- problemas envolvendo um conjunto finito de itens cuja estrutura associada não é clara. Esta é modelada como um conjunto de nós e arcos *fuzzy*, denominado *grafo fuzzy*.

Para um melhor entendimento dos problemas abordados nesta tese, os dois problemas de grafos supracitados são apresentados neste capítulo. Ressalta-se que a maioria dos problemas tratados nesta tese utilizam o primeiro tipo (estrutura *crisp* e parâmetros *fuzzy*).

Este capítulo está organizado da seguinte forma: na Seção 2 é apresentado o problema em grafos com estrutura *crisp* e parâmetros *fuzzy*. Na Seção 3 é abordado o problema com estrutura de grafos *fuzzy*. Considerações finais estão na Seção 4.

3.2 Problemas com estrutura do grafo *crisp* e parâmetros *fuzzy*

Este é o problema mais citado na literatura. Trata-se de um problema onde a estrutura do grafo é bem conhecida e seus parâmetros associados são incertos, representados por números *fuzzy*. São vários os exemplos, principalmente considerando custo ou tempo como números *fuzzy* (vide Figura 3.1).

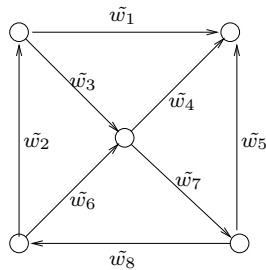


Figura 3.1: Exemplo de grafo com parâmetros *fuzzy*

Outros problemas clássicos, nos quais as incertezas estão associadas aos parâmetros são os problemas de fluxo em redes. No problema de fluxo máximo é associado um grau de confiabilidade da solução dado o fluxo nos arcos (vide Capítulo 5). Este problema foi bastante estudado por Chanas (1987). Um problema mais complexo que envolve parâmetros *fuzzy*, tanto na capacidade dos arcos quanto nos custos associados, é o problema de fluxo de custo mínimo, que é objeto de estudo do Capítulo 6.

Quanto à solução, de um modo geral, a proposta por Bellman e Zadeh (1970) é a mais utilizada para o problema de grafos com parâmetros *fuzzy*, sendo definida na próxima seção. Também é apresentada a abordagem adotada por Okada (2004) para o problema de caminho mínimo com parâmetros *fuzzy*, visto que é referência aos algoritmos propostos no Capítulo 4.

3.2.1 Abordagem de Bellman e Zadeh, 1970

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

Definição 20 *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:*

$$D = C \cap R \quad (3.1)$$

e $\mu_D = \mu_C \wedge \mu_R$ (Figura 3.2).

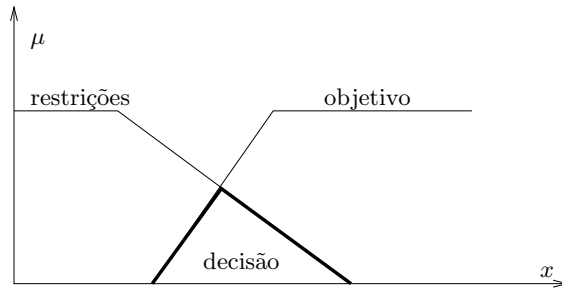


Figura 3.2: Exemplo de um gráfico do modelo de Bellman e Zadeh

Portanto, considerando uma decisão *ótima*, tem-se:

$$\mu_{D_{\max}} = \max_{x_v} \{ \mu_C(x_v) \wedge \mu_R(x_v) \} \quad (3.2)$$

tal que x_v é uma solução factível para o problema em grafos. Na Figura 3.2, 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.

3.2.2 Abordagem de Okada, 2004

Seja, por exemplo, o problema de caminho mínimo com parâmetros *fuzzy* (Capítulo 4), isto é, em um grafo G com custos *fuzzy* $\tilde{c}_{ij}$. Este problema tem como objetivo encontrar o menor caminho entre dois nós da rede. Seja o exemplo da Figura 3.3:

tal que os custos são números triangulares *fuzzy* representados na forma (m, α, β) .

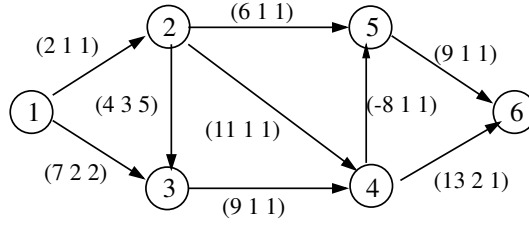


Figura 3.3: Exemplo - caminho mínimo

Por exemplo, analisando os possíveis caminhos entre os nós 1 e 6 (Figura 3.3) foram encontrados, dentre os vários existentes, três possíveis caminhos com os seguintes custos: $\tilde{a} = (14, 4, 4)$, $\tilde{b} = (17, 3, 3)$, $\tilde{c} = (29, 5, 4)$ (Figura 3.4).

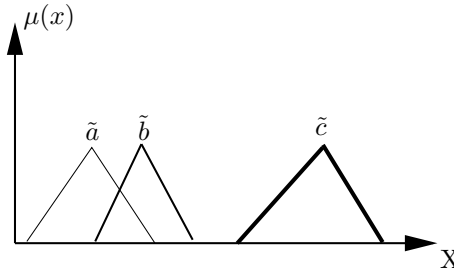


Figura 3.4: Caminhos do grafo da Figura 3.3

É fácil ver que o custo *fuzzy* associado ao caminho $\tilde{c}$ (Figura 3.4) é maior do que o custo dos demais caminhos, o qual pode ser descartado. Porém, os outros dois caminhos têm um considerável *valor de possibilidade* (usando a definição de Okada (2004)) de ser o menor caminho.

Muitas vezes não é interessante determinar como caminho mínimo somente o caminho com maior grau de possibilidade, dada alguma ordenação, o que poderia ser visto como perda de informação. Um conjunto solução, onde cada caminho possui um grau de pertinência associado, pode ser muito proveitoso ao decisor.

Com base neste fato, alguns algoritmos (Okada e Soper 2000, Okada 2004) para problemas de caminho mínimo *fuzzy* foram construídos de modo a fornecer todo o conjunto solução para o problema ($\mathcal{P} = \{p_{\mu_i}^i, i = 1, \dots, n\}$, ou seja, um caminho p^i com grau de pertinência μ_i ao conjunto solução). Também foi construído um algoritmo para o caso de fluxo máximo com estrutura e solução *crisp*, mas com capacidade *fuzzy*. A construção destes algoritmos é intuitivamente simples, mas com dificuldades intrínsecas que são abordadas a seguir.

Chanas (1987), Okada e Soper (2000), Okada (2004) utilizaram a teoria de possibilidade (Zadeh 1978) para calcular a pertinência de determinada solução para o problema de caminho mínimo. Blue, Bush e Puckett (2002) também utilizaram esta teoria, embora não tenham explicitado isso em seu trabalho. Para construir a solução deste problema tem-se que:

1. encontrar todos os caminhos candidatos; e
2. calcular o valor de possibilidade entre as soluções encontradas.

Analisando os itens supracitados, verifica-se que o item 1 é um problema combinatorial. Após resolvido este problema, ainda é necessário 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, pode-se considerar formas de armazenamento das informações, dentre outros detalhes que facilitariam a resolução do problema. Com base nisto, são propostas, nesta tese, heurísticas para abordar estes problemas. Tais problemas são estudados nos capítulos subsequentes.

3.3 Problemas com estrutura de grafos *fuzzy*

Neste caso o grafo $\tilde{G} = (\tilde{V}, \tilde{A})$ não é bem conhecido, tendo valores de pertinência associados aos seus arcos $(i, j) \in \tilde{A}$ e/ou aos seus nós $i \in \tilde{V}$, como descrito na definição de Rosenfeld (1975). No Capítulo 7 desta tese são estudadas algumas aplicações de grafos *fuzzy*.

Para este tipo de problema, o trabalho de Delgado, Verdegay e Vila (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, Delgado, Verdegay e Vila (1995) para o problema do fluxo ótimo em estrutura de grafos *fuzzy*,

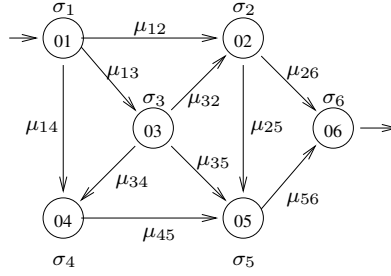


Figura 3.5: Exemplo de um grafo com estrutura *fuzzy*

que foi utilizado para resolver um problema de transporte. Na seção seguinte, o algoritmo fundamental de Delgado et al. (1990) é apresentado.

3.3.1 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 V e μ é a função de pertinência associada ao conjunto de arcos A , sendo que o valor de pertinência de um arco (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 $G^\alpha = (V^\alpha, A^\alpha)$. O grafo resultante é um grafo *crisp* tal que:

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

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

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

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

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}

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

como função custo.

No artigo, considera-se a função custo c_{ij} associada ao grafo dependente do custo 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

$$\tau_i = \max_{j \in V} \{\nu_{ij} \bigvee \nu_{ji}\}$$

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* é utilizado no capítulo sobre coloração *fuzzy* desta tese.

3.4 Considerações finais

Neste capítulo foram apresentados os principais problemas que envolvem grafos *fuzzy*, sendo os problemas com incertezas nos parâmetros e/ou na estrutura. Foram também apresentados os principais trabalhos da literatura para cada um destes problemas (Bellman e Zadeh, Okada, Delgado et al).

Capítulo 4

O Problema de Caminho Mínimo com Incertezas

4.1 Introdução

O problema de encontrar um caminho mínimo de um nó origem específico a outros nós de uma rede é um dos mais estudados da teoria dos grafos, pois aparece em muitas aplicações, tais como: transporte, roteamento, telecomunicações, etc.

Seja $G = (V, A)$ um grafo orientado, onde V é o conjunto de nós e A o conjunto de arcos. Um *caminho* entre dois nós é uma seqüência alternada de nós e arcos possuindo como pontos iniciais e finais um nó. Logo, o *comprimento* (custo) de um caminho é a soma dos pesos (custos) dos arcos pertencentes a este caminho. Na maioria dos casos existe mais de um caminho entre dois nós específicos, com isso o *problema de caminho mínimo* consiste em encontrar o caminho com menor custo dentre todos os possíveis.

Em se tratando deste problema, na teoria dos grafos clássica, o peso (custo) de cada arco é um número *crisp*, sendo que, para este caso, existem algoritmos eficientes para resolvê-los. Porém, em muitas aplicações reais estes parâmetros (custos, capacidades, demandas, dentre outros) não são naturalmente precisos, muitas vezes por falta de informações. Embora este assunto tenha um vasto campo de aplicações, o estudo nesta área, considerando incertezas nos parâmetros, ainda se encontra na fase inicial. Além do mais, a maioria dos problemas propostos são NP-completos, o que torna difícil suas soluções. Portanto, neste capítulo é abordado o problema de caminho mínimo em grafos com parâmetros incertos, sendo que

as incertezas são tratadas com o auxílio da teoria dos conjuntos *fuzzy*, por isso também é denotado de *problema de caminho mínimo fuzzy*.

Na literatura existem alguns trabalhos abordando este tema, que são apresentados na próxima seção. Porém, verificou-se que estes são aplicados somente em redes que possuem custos não-negativos. Além disso, apresentam particularidades que merecem ser estudadas, tais como: alguns algoritmos encontram custos finais sem um caminho associado (maiores detalhes: Takahashi (2004)), outros apresentam um conjunto solução de caminhos sem maiores informações ao usuário sobre qual escolher. Portanto, um dos objetivos do algoritmo proposto neste capítulo é contornar estas dificuldades.

Tal algoritmo é baseado no algoritmo clássico de Ford-Moore-Bellman (Bellman 1958), podendo ser aplicado em redes com parâmetros negativos, além de detectar a existência de circuito negativo. Em algumas aplicações reais (problemas de redes de comunicações, transportes, dentre outras) a existência de custos negativos é freqüente (maiores detalhes: Ahuja et al. (1993)), sendo que na literatura não há algoritmos para estes tipos de aplicações, sendo esta mais uma justificativa para o algoritmo proposto neste capítulo.

Este capítulo está organizado da seguinte forma: na Seção 2 são comentados os principais trabalhos da literatura. A formulação matemática do problema está na Seção 3. A Seção 4 apresenta o algoritmo proposto. A inclusão de restrições de tempo no algoritmo proposto está na Seção 5. Na Seção 6 é adaptada uma heurística para ordenar os caminhos não-dominados encontrados pelo algoritmo estudado. Uma relação de ordem genérica para o mesmo é abordada na Seção 7. A Seção 8 apresenta as considerações finais.

4.2 Revisão bibliográfica

O problema de caminho mínimo em grafos com incertezas é um problema de alta complexidade computacional, visto que os custos finais dos caminhos são números *fuzzy* e a comparação entre estes é difícil, sendo às vezes impossível determinar o menor caminho. Devido a isso, alguns trabalhos utilizam os índices de defuzzificação existentes na literatura para assim resolver um problema clássico (*crisp*), utilizando os algoritmos já existentes. Outros apre-

sentam um conjunto solução de caminhos, deixando a cargo do usuário escolher o caminho que melhor lhe adaptar.

Dentre os principais trabalhos da literatura destacam-se: Dubois e Prade (1980), Klein (1991), Lin e Chern (1993), Chang e Lee (1999), Okada e Soper (2000), Blue et al. (2002), Okada (2004), Chuang e Kung (2005) e Nayeem e Pal (2005).

Um dos primeiros trabalhos envolvendo este tema se deve a Dubois e Prade (1980), o qual propõe as extensões dos algoritmos clássicos de Floyd e de Ford-Moore-Bellman. Porém, verificou-se que ambos os algoritmos retornam comprimentos (custos) sem um caminho associado (maiores detalhes: Takahashi (2004)), problema este que foi contornado por Klein (1991) com o uso de dominância de conjuntos *fuzzy*.

Lin e Chern (1993) definem a denominação de arcos vitais, como sendo aqueles cuja remoção da rede resulta em um aumento da distância mínima.

No trabalho de Chang e Lee (1999) é utilizado um método de ordenação de números *fuzzy* que associa a cada número *fuzzy* um valor *crisp* e com este valor resolvem um problema clássico associado.

Okada e Soper (2000) caracterizam a solução do problema não como o caminho mínimo com maior grau de pertinência possível, mas como um conjunto solução *fuzzy*, onde cada elemento é um caminho não-dominado. Nesse trabalho foi definida uma relação de ordem entre números *fuzzy* e foi introduzido o conceito de dominância entre caminhos, conceitos estes que são utilizados no algoritmo proposto.

Blue et al. (2002) apresentam uma taxonomia para grafos e propõem alguns algoritmos para os principais problemas. No algoritmo de caminho mínimo proposto, os autores encontram um valor de corte para limitar o número de caminhos a serem analisados, em seguida aplica-se uma versão modificada do algoritmo dos k-caminhos mínimos (*crisp*), proposto por Eppstein (1994), a fim de se determinar os caminhos com algum grau de possibilidade de ser o mínimo.

Seguindo a idéia de encontrar um conjunto solução, Okada (2004) analisou a interatividade entre caminhos *fuzzy* e introduziu o conceito de grau de possibilidade de um arco pertencer

a um caminho mínimo.

Chuang e Kung (2005) apresentam uma heurística que encontra um número *fuzzy* que tenha o menor limitante inferior, modal e limitante superior entre todos os caminhos existentes na rede, número este chamado de *custo do caminho mínimo fuzzy* (CCMF), denotado também de L_{min} . Após encontrado o CCMF, foi proposta a definição de grau de similaridade, cuja finalidade é encontrar o caminho, dentre todos os possíveis caminhos, que esteja mais próximo possível do CCMF (L_{min}).

Figurando entre um dos mais recentes trabalhos da literatura, Nayeem e Pal (2005) propõem um algoritmo, utilizando o índice de aceitabilidade proposto por Sengupta e Pal (2000), onde o usuário, de acordo com seu ponto de vista (otimista/pessimista), deve escolher o melhor caminho. Porém esta escolha só ocorre quando, na solução final, houver mais de um caminho entre dois nós, sendo que isso somente acontecerá quando mais de um caminho possuir os mesmos valores modais, pois, caso contrário sempre haverá um caminho mínimo.

Como dito anteriormente, os trabalhos comentados nesta seção possuem a desvantagem de lidarem somente com parâmetros positivos e por isso, é proposto um algoritmo que contorna esta dificuldade.

4.3 Formulação matemática

Considere uma rede direcionada $G = (V, A)$, tal que: V é o conjunto dos nós; A o conjunto dos arcos, onde cada arco é denotado por um par ordenado $(i, j) \in A$, sendo $i, j \in V$. Supondo que há somente um arco direcionado (i, j) de i para j e assumindo 1 como o nó origem e t o nó destino, então um caminho p_{1t} é definido como uma sequência alternada de nós e arcos ($p_{1j} = \{1, (1, i_1), i_1, \dots, i_k, (i_k, t), t\}$) e é assumido também que há pelo menos um caminho p_{1t} em $G = (V, A)$ entre os nós 1 e t , $t \in V - \{1\}$. Logo, o *problema de caminho mínimo fuzzy* é formulado da seguinte forma (Okada 2004):

$$\begin{aligned}
\min \tilde{f}(x) &= \sum_{(i,j) \in A} \tilde{c}_{ij} x_{ij} \\
s.a. : \quad &\sum_{j:(i,j) \in A} x_{ij} - \sum_{k:(k,i) \in A} x_{ki} = \begin{cases} 1, & \text{se } i = 1 \\ 0, & \text{se } i \neq 1, t \text{ (} i = 1, \dots, r \text{)} \\ -1, & \text{se } i = t \end{cases} \\
&x_{ij} = 0 \text{ ou } 1 \text{ para } (i, j) \in A
\end{aligned} \tag{4.1}$$

tal que: r é o número de nós e o símbolo $\sum$ na função objetivo refere-se a adição $\oplus$ de números *fuzzy*.

O significado de "*min*" na Equação (4.1) é ambíguo, pois depende do índice de ordenação de números *fuzzy* utilizado, sendo que a solução ótima pode não ser obtida.

4.4 Algoritmo proposto

Nesta seção é apresentado o algoritmo proposto, bem como a relação de ordem, proposta por Okada e Soper (2000), utilizada para determinar o conjunto de caminhos não-dominados.

4.4.1 Relação de ordem de Okada e Soper

Em Okada (2004) constatou-se que um algoritmo que encontra o conjunto *fuzzy* solução com todos os caminhos que possuem algum grau de possibilidade de ser a solução ótima é do tipo NP-completo pois, além de proceder a enumeração dos caminhos entre dois nós, estes caminhos devem ser comparados para determinação do grau de possibilidade. Além disso, estes procedimentos sofrem a influência da incerteza e da dimensão do problema a ser resolvido. Por estes fatores, um algoritmo que construa um subconjunto representativo do conjunto *fuzzy* de soluções, seria mais atraente para o decisor. Por isso, foi utilizado no algoritmo proposto, para a construção do conjunto solução, o conceito de dominância de caminhos de Okada e Soper (2000), definida na Seção 2.7 do Capítulo 2.

4.4.2 Apresentação do algoritmo

Como citado anteriormente, o algoritmo proposto é uma adaptação do algoritmo clássico de Ford-Moore-Bellman (Bellman 1958). Este é um algoritmo iterativo, possuindo como critério de parada o número de iterações ou a não alteração dos custos de todos os caminhos

encontrados da iteração anterior com relação à iteração atual. Como a relação de Okada e Soper pode apresentar, entre dois nós, mais de um caminho não-dominado, então cada caminho recebe um rótulo (etiqueta) para que este seja construído no passo final do algoritmo. Tal algoritmo é baseado nos seguintes passos:

a) No Passo 1 é feita a inicialização das variáveis, onde são atribuídos os valores (rótulos e custos) iniciais dos caminhos (iteração "zero"), para assim após a primeira iteração (Passo 2) verificar o critério de parada.

b) O Passo 2 é composto de duas importantes etapas. Na primeira (Item 2 do Passo 2) são determinados todos os caminhos entre os nós 1 e i (são determinados através das distâncias associadas no Passo 1 e com os custos dos arcos (i, j) ($\tilde{l}_{ij}$). Na segunda etapa (Item 3 do Passo 2) é aplicada a relação de dominância de Okada e Soper para descartar os caminhos dominados.

c) No Passo 3 é verificado o critério de parada. Se este é satisfeito, os caminhos não-dominados são construídos no Passo 4 e o algoritmo é finalizado no Passo 5. Caso contrário, o algoritmo retorna ao Passo 2 (como o algoritmo analisa todos os nós a cada iteração, similar ao de Ford-Moore-Bellman, as distâncias (custos) entre alguns caminhos podem alterar seus valores) ou finaliza caso exista circuito negativo.

Informações sobre o algoritmo proposto

V : conjunto dos nós;

it : contador de iterações;

$(m + \beta)^i$: espalhamento à direita $(m + \beta)$ do custo do nó i (limitante superior);

$\tilde{l}_{ji}$: custo do arco (j, i) ;

$\tilde{c}_{(i,k)}^{it}$: custo do caminho entre os nós 1 e i , com a etiqueta k , na iteração it ;

M : um número com valor grande, substitui o ∞ do algoritmo clássico de Ford-Moore-Bellman; e

Γ_i^{-1} : conjunto dos nós predecessores de i .

Algoritmo

PASSO 1: Inicialização das variáveis

1. $\tilde{c}_{(1,1)}^0 = (0, 0, 0)$
2. $\tilde{c}_{(j,1)}^0 = (M + 2, 1, 1), j = 2, 3, \dots, r$

tal que:

- r : números de nós;
- $M = \sum_{i=1}^{na} |(m + \beta)^i|$; e
- na : número de arcos.

3. $it \leftarrow 1$.

PASSO 2: Determinação dos caminhos e verificação da dominância

1. $\tilde{c}_{(1,1)}^{it} = (0, 0, 0)$
2. $\forall j \in \Gamma_i^{-1}, i = 2, 3, \dots, r$, faça:
 - $\tilde{c}_{(i,k1)}^{it} = \tilde{c}_{(j,k2)}^{it-1} \oplus \tilde{l}_{ji}$ (construção dos custos dos caminhos)

3. Verificação da dominância entre as etiquetas do nó i (relação de Okada e Soper):

- Se $\tilde{c}_{(i,m)}^{it} \succ \tilde{c}_{(i,n)}^{it} \Rightarrow$ elimine a m -ésima etiqueta
- Se $\tilde{c}_{(i,m)}^{it} \prec \tilde{c}_{(i,n)}^{it} \Rightarrow$ elimine a n -ésima etiqueta

PASSO 3: Critério de parada

1. Se $(\tilde{c}_{(i,k1)}^{it} = \tilde{c}_{(i,k1)}^{it-1}, \forall i \in V)$ ou $(it = r)$ faça:
 - Se $it = r$ e $\tilde{c}_{(i,k1)}^{it} \neq \tilde{c}_{(i,k1)}^{it-1} \Rightarrow$ Passo 5 (circuito negativo)
 - Senão $\Rightarrow$ Passo 4

2. Senão: $it \leftarrow it + 1 \Rightarrow$ volte ao Passo 2

PASSO 4: Composição dos caminhos

- Encontre todos os caminhos não-dominados entre os nós 1 e i

PASSO 5: FIM.

Vale ressaltar que o algoritmo foi proposto para números triangulares, porém a extensão para trapezoidais é simples e imediata.

4.4.3 Estudo da complexidade e da convergência

4.4.3.1 Comentários gerais

Como citado em Goldbarg e Luna (2000), o algoritmo de Ford-Moore-Bellman (FMB) não analisa um único nó a cada iteração, mas sim, examina todos os nós até que melhorias não mais sejam possíveis, podendo assim aceitar arcos com custos negativos. Dado que o algoritmo proposto é uma adaptação do algoritmo de FMB, a estrutura é mantida.

4.4.3.2 Estudo da otimalidade

O algoritmo proposto busca um conjunto de soluções, sendo que cada solução possui um grau de otimalidade. O princípio de otimalidade utilizado é baseado no princípio de otimalidade de Bellman, que pode ser encontrado em Gondran e Minoux (1984).

4.4.3.3 Complexidade e convergência

Dado que o algoritmo de Ford-Moore-Bellman, em caso de inexistência de um circuito negativo, converge em no máximo $r-1$ iterações (r : número de nós), este algoritmo também irá convergir em no máximo $r-1$ iterações, visto que a análise de convergência dos dois algoritmos é a mesma.

Quanto ao estudo da complexidade deste, tem-se que no Passo 1 são efetuadas no máximo rV_{max} somas para calcular o custo de cada caminho, sendo r o número de nós e V_{max} o máximo entre $\{V_1, V_2, \dots, V_r\}$, onde V_j é o número de rótulos (etiquetas) permanentes do nó j . Também tem-se no máximo rV_{max}^2 comparações para dominância. Logo, para cada iteração tem-se uma complexidade de $O(rV_{max} + rV_{max}^2) = O(rV_{max}^2)$. Dado que o algoritmo executa no máximo $r-1$ iterações, a complexidade se torna $O((r-1)(rV_{max}^2)) = O(r^2V_{max}^2 - rV_{max}^2) = O(r^2V_{max}^2)$.

4.4.4 Resultados computacionais

Como o algoritmo proposto possui a relação de ordem de Okada e Soper (2000), os dois algoritmos foram implementados com a finalidade de compará-los, na linguagem Matlab 7.0 e testados em uma rede óptica italiana de telecomunicações (Ali, Ramamurthy e Deogun 2003) e em uma outra pequena rede com parâmetros negativos. Demais exemplos podem ser encontrados em Hernandez, Lamata, Takahashi, Verdegay e Yamakami (2006), Hernandez, Lamata, Verdegay e Yamakami (2006) e Hernandez e Yamakami (2004).

Exemplo 4.1 Rede italiana

Neste exemplo foi considerada a rede acíclica da Figura 4.1, com todos os arcos positivos, sendo que a numeração dos mesmos estão em ordem crescente, conforme Tabela 4.2, ou seja, os arcos (i,j) estão ordenados na forma $i < j$.

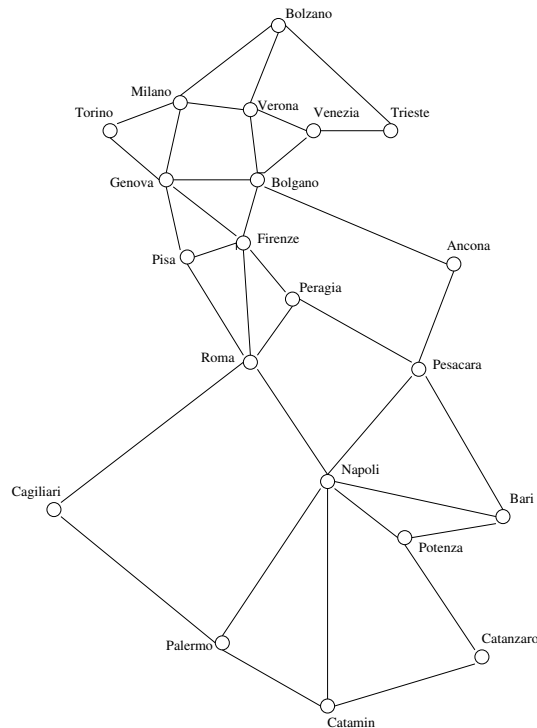


Figura 4.1: Rede óptica italiana

As Tabelas 4.1 e 4.2 representam, respectivamente, os nós e os custos dos arcos da rede da Figura 4.1.

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

Tabela 4.1: Denominação dos nós da rede italiana

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

Tabela 4.2: Tabela dos arcos e custos da rede italiana

Após executados os resultados, percebe-se que para ambos os algoritmos os resultados foram idênticos. Isto aconteceu porque a rede obedece aos requisitos exigidos no algoritmo de Okada e Soper, além do mais, esta possui espalhamentos à direita e à esquerda idênticos.

Para ilustrar: o único caminho não-dominado do nó 1 ao nó 21 é:

$$1 \rightarrow 3 \rightarrow 8 \rightarrow 9 \rightarrow 13 \rightarrow 15 \rightarrow 21 \text{ com custo } (970 \ 88 \ 88).$$

Exemplo 4.2 Rede acíclica.

Neste exemplo a rede considerada (Figura 4.2) é acíclica e apresenta um arco com custo negativo.

Executando os dois algoritmos, para a rede do Exemplo 4.2, e verificando os caminhos não dominados do nó 1 ao nó 5 foram obtidas duas situações:

1. O algoritmo de Okada e Soper apresentou três caminhos não-dominados:

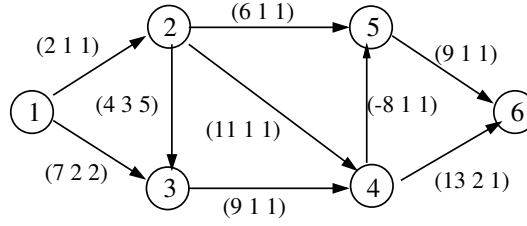


Figura 4.2: Rede com custos negativos

- $1 \rightarrow 2 \rightarrow 5$ com custo (8 2 2)
- $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ com custo (7 6 8)
- $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$ com custo (5 3 3)

2. O algoritmo proposto apresentou apenas dois caminhos não-dominados:

- $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ com custo (7 6 8)
- $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$ com custo (5 3 3)

Analisando os resultados percebe-se que o primeiro algoritmo (Okada e Soper) apresentou o primeiro caminho como não-dominado, o que de fato não ocorre, pois este é dominado pelo terceiro. Porém, isso já era de se esperar, pois este algoritmo foi desenvolvido somente para redes com arcos de custos positivos. Quanto ao algoritmo proposto verifica-se que este encontra de fato os caminhos não dominados do nó 1 ao nó 5. Isto ocorreu porque este foi desenvolvido para redes com custos nos arcos positivos e também negativos (maiores detalhes: Hernandez e Yamakami (2004)).

4.4.5 Conclusões da Seção 4.4

Verificando os resultados dos exemplos anteriores verificou-se que este algoritmo além de possuir as vantagens já citadas (pode ser aplicado em redes com custos quaisquer; detecta a existência de circuito negativo; possui uma complexidade computacional ($O(r^2 V_{max}^2)$) menor do que a do algoritmo de Okada e Soper ($O(r^3 V_{max}^2)$), e suporta a inclusão de mais algumas restrições, a saber: a inclusão de restrições de tempo nos nós; a inclusão da heurística de (Chuang e Kung 2005), para detectar o menor caminho entre os não-dominados; e, além disso, é possível a generalização da relação de ordem. Com base nisto, nas próximas seções deste capítulo serão incluídas estas restrições.

4.5 Problema de caminho mínimo com incertezas e restrições de tempo

Nesta seção é apresentada uma adaptação do algoritmo proposto (Seção 4.4.2), onde foi incluída uma restrição de tempo em cada um dos nós da rede, ou seja, o objetivo do novo algoritmo é de encontrar todos os caminhos não-dominados que satisfazem as restrições de tempo entre os nós, restrição esta que também é um número *fuzzy*.

4.5.1 Restrições de tempo

Neste problema a rede a ser considerada possui dois parâmetros incertos nos arcos, sendo um o custo e outro o tempo. Logo, cada caminho entre o nó 1 e i possui custo e tempo incertos. Além destes dois parâmetros referentes aos caminhos, em cada nó i da rede há uma restrição de tempo, ou seja, os caminhos entre os nós 1 (inicial) e i devem satisfazer a restrição de tempo daquele nó i . Esta restrição é o tempo máximo permitido para um caminho entre dois nós, mais especificamente entre o nó 1 e o nó i . A definição é satisfeita da seguinte forma:

Definição 21 *Seja $\tilde{T}_{(i,k)}^{it}$ o tempo do caminho não dominado entre o nó 1 e i (p_{1i}) com a etiqueta k e iteração it , e seja $\tilde{T}_i$ a restrição de tempo do nó i , então p_{1i} satisfaz a restrição de tempo se $Poss(\tilde{T}_{(i,k)}^{it} \leq \tilde{T}_i) > 0$.*

4.5.2 Algoritmo com restrição de tempo

A adaptação do algoritmo da Seção 4.4.2 ocorre no Passo 2 do mesmo, sendo os outros passos idênticos. Além de verificar todos os caminhos entre os nós 1 e i que satisfazem as restrições de tempo do respectivo nó i , no Item 6 do Passo 2 é encontrado o caminho que possui o menor valor modal e a partir daí os demais caminhos são ordenados através do cálculo da possibilidade.

Informações sobre o algoritmo proposto

V : conjunto dos nós;

it : contador de iterações;

$(m + \beta)^i$: limitante superior do nó i ;

$\tilde{l}_{ji}$: custo do arco (j, i) ;

$\tilde{t}_{ji}$: tempo do arco (j, i) ;

$\tilde{c}_{(i,k)}^{it}$: custo do caminho entre os nós 1 e i com a etiqueta k , na iteração it ;

$\tilde{T}_{(i,k)}^{it}$: tempo total do caminho entre os nós 1 e i com a etiqueta k , na iteração it ;

$\tilde{T}_i$: restrição de tempo do nó i ;

$\tilde{p}_{(1,i)}^{it}$: caminho do nó 1 ao i na iteração it ;

$\tilde{m}_{(1,i)}^{it}$: valor modal do caminho do nó 1 ao i na iteração it ;

M : um número com valor grande, substitui o ∞ do algoritmo clássico de Ford-Moore-Bellman; e

Γ_i^{-1} : conjunto dos nós predecessores de i .

Algoritmo com restrição de tempo

PASSO 1: Inicialização das variáveis

1. $\tilde{c}_{(1,1)}^0 = (0, 0, 0)$
2. $\tilde{T}_{(1,1)}^0 = (0, 0, 0)$
3. $\tilde{c}_{(j,1)}^0 = (M + 2, 1, 1)$, $j = 2, 3, \dots, r$

tal que:

- r : números de nós;
- $M = \sum_{i=1}^{na} |(m + \beta)^i|$; e
- na : número de arcos.

4. $it \leftarrow 1$

PASSO 2: Determinação dos caminhos e verificação da dominância

1. $\tilde{c}_{(1,1)}^{it} = (0, 0, 0)$

2. $\tilde{T}_{(1,1)}^{it} = (0, 0, 0)$
3. $\forall j \in \Gamma_i^{-1}, i = 2, 3, \dots, r$, faça:
 - $\tilde{c}_{(i,k1)}^{it} = \tilde{c}_{(j,k2)}^{it-1} \oplus \tilde{l}_{ji}$
 - $\tilde{T}_{(i,k1)}^{it} = \tilde{T}_{(j,k2)}^{it-1} \oplus \tilde{t}_{ji}$
4. Verificação da dominância entre as etiquetas do nó i :
 - Se $\tilde{c}_{(i,m)}^{it} \succ \tilde{c}_{(i,n)}^{it} \Rightarrow$ elimine a m -ésima etiqueta
 - Se $\tilde{c}_{(i,m)}^{it} \prec \tilde{c}_{(i,n)}^{it} \Rightarrow$ elimine a n -ésima etiqueta
5. Para todos os caminhos não-dominados ($p_{(1,i)}^{it}$) entre o nó 1 e o nó i calcule:

$$\tilde{a} = Poss(\tilde{T}_{(1,i)}^{it} \leq \tilde{T}_i)$$
 (denotada, na tabela de resultados, por $Poss(\text{tempo})$)
 - Se $\tilde{a} = 0$ elimine o caminho $p_{(1,i)}^{it}$
6. Entre os caminhos não-dominados do nó 1 ao i que satisfazem as restrições de tempo faça:
 - Verifique o caminho que possui o menor valor modal ($\tilde{m}_{(1,i)}^{it}$)
 - calcule $Poss(\tilde{c}_{(i,k)}^{it} \leq \tilde{m}_{(1,i)}^{it})$ (denotada, na tabela de resultados, por $Poss(\text{custo})$)

PASSO 3: Critério de parada

1. Se $(\tilde{c}_{(i,k1)}^{it} = \tilde{c}_{(i,k1)}^{it-1} \forall i \in V)$ ou $(it = r)$ faça:
 - Se $it = r$ e $\tilde{c}_{(i,k1)}^{it} \neq \tilde{c}_{(i,k1)}^{it-1} \Rightarrow$ Passo 5 (circuito negativo)
 - Senão $\Rightarrow$ Passo 4
2. Senão: $it \leftarrow it + 1 \Rightarrow$ volte ao Passo 2

PASSO 4: Composição dos caminhos

- Encontre todos os caminhos não-dominados entre os nós 1 e i

PASSO 5: FIM.

Ressalta-se que o limitante inferior das restrições de tempo adotado foi zero ($\tilde{0} = (0, 0, 0)$).

4.5.3 Resultados computacionais

O algoritmo proposto foi implementado na linguagem Matlab 7.0 e testada na seguinte rede óptica europeia COST239 da Figura 4.3 (Tan e Sinclair 1995):

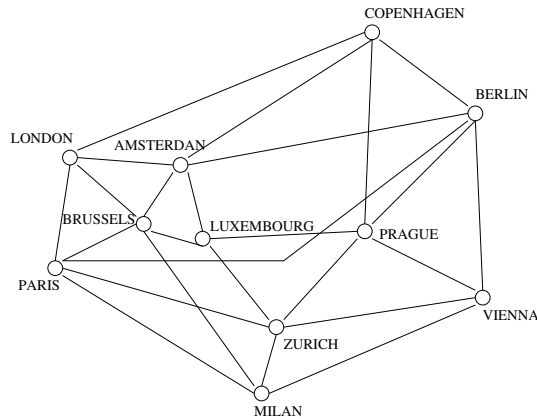


Figura 4.3: Rede europeia - COST239

Os nós da rede e suas restrições de tempo estão definidos na Tabela 4.3.

Nós	Cidades	Restrições-Tempo
1	Paris	(0 0 0)
2	Milan	(1,7 0,1 0,3)
3	Zurich	(1 0,1 0,3)
4	Prague	(1,7 0,3 0,2)
5	Vienna	(2,2 0,2 0,3)
6	Berlin	(1 0,2 0,1)

Nós	Cidades	Restrições-Tempo
7	Amsterdan	(0,8 0,1 0,1)
8	Luxembourg	(1 0,1 0,1)
9	Brussels	(0,45 0,15 0,05)
10	London	(1,2 0,2 0,1)
11	Copenhagen	(1,95 0,3 0,05)

Tabela 4.3: Denominação dos nós da rede (COST239) e restrições de tempo

Na Tabela 4.4 estão os nós origem e destino de cada arco, bem como os seus respectivos custos e tempos, enquanto que na Tabela 4.5 são apresentados os resultados do algoritmo adaptado.

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

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

Arco	Origem	Destino	Custos	Tempos
1	1	2	(820 20 20)	(1,7 0,2 0,2)
2	1	3	(361 11 9)	(0,9 0,1 0,1)
3	1	6	(677 27 6)	(1,05 0,1 0,1)
4	1	9	(300 10 50)	(0,52 0,12 0,08)
5	1	10	(450 30 20)	(1 0,05 0,3)
6	2	3	(186 6 7)	(0,45 0,1 0,05)
7	2	5	(510 15 15)	(1,5 0,25 0,4)
8	2	9	(930 30 30)	(2 0,3 0,1)
9	3	4	(667 17 196)	(0,98 0,08 0,22)
10	3	5	(748 18 22)	(1,3 0,15 0,15)
11	3	8	(443 18 22)	(0,6 0,1 0,1)
12	4	5	(199 9 11)	(0,3 0,1 0,1)
13	4	6	(340 30 20)	(0,65 0,15 0,05)
14	4	11	(740 30 30)	(1,15 0,1 0,05)
15	5	6	(660 50 30)	(0,85 0,1 0,11)
16	6	11	(242 12 18)	(0,9 0,1 0,1)
17	7	6	(410 20 30)	(0,85 0,1 0,11)
18	7	11	(472 22 18)	(1,16 0,16 0,14)
19	8	4	(730 20 5)	(1,2 0,15 0,05)
20	8	7	(242 12 13)	(0,42 0,12 0,11)
21	9	8	(137 7 8)	(0,3 0,05 0,05)
22	9	7	(130 10 20)	(0,28 0,03 0,04)
23	9	10	(242 12 18)	(0,42 0,12 0,11)
24	10	7	(342 12 8)	(0,65 0,15 0,05)
25	10	11	(1310 60 120)	(2,45 0,15 0,15)

Tabela 4.4: Tabela dos dados da rede COST239

Origem	Destino	Caminho	Custo Caminho	Tempo Caminho	Poss(custo)	Poss(tempo)
1	2	1 → 2	(820 20 20)	(1,7 0,2 0,2)	1,0	1,0
1	3	1 → 3	(361 11 9)	(0,9 0,1 0,1)	1,0	0,5
1	4	1 → 3 → 4	(1028 28 205)	(1,88 0,18 0,32)	1,0	0,5263
1	4	1 → 9 → 8 → 4	(1167 37 63)	(2,02 0,32 0,18)	0,4256	0,3846
1	5	1 → 3 → 5	(1109 29 31)	(2,2 0,25 0,25)	1,0	1,0
1	6	1 → 6	(677 27 6)	(1,05 0,1 0,1)	1,0	0,75
1	7	1 → 9 → 7	(430 20 70)	(0,8 0,15 0,12)	1,0	1,0
1	8	1 → 9 → 8	(437 17 58)	(0,82 0,17 0,13)	1,0	0,2174
1	9	1 → 9	(300 10 50)	(0,52 0,12 0,08)	1,0	0,5882
1	10	1 → 10	(450 30 20)	(1 0,05 0,3)	1,0	0,6
1	11	1 → 6 → 11	(919 39 24)	(1,95 0,2 0,2)	0,8661	0,8333
1	11	1 → 9 → 7 → 11	(902 42 88)	(1,96 0,31 0,24)	1,0	1,0

Tabela 4.5: Resultados da rede COST239

(1 → 9 → 8 → 4). Isto significa que o primeiro caminho satisfaz melhor as condições de otimalidade e factibilidade do que o segundo.

Seja, por exemplo, o caminho 1 → 2 → 3 → 4, sendo seu custo total (1673 43 223) e seu tempo (3,13 0,38 0,47). Analisando o custo, verifica-se que este caminho é dominado pelos custos dos outros caminhos da Tabela 4.5 (1028 28 205; 1167 37 63). Além disso, o tempo da restrição do nó 4 (Tabela 4.3) não é satisfeito, isto é, a medida de possibilidade é nula. Portanto, o caminho 1 → 2 → 3 → 4 não é considerado no conjunto solução.

A análise dos caminhos entre os nós 1 e 11 é semelhante à feita em relação aos caminhos entre os nós 1 e 4.

Percebe-se também que neste exemplo é possível aplicar um corte no conjunto solução,

ou seja, os caminhos que possuírem a medida de possibilidade menor do que um valor estabelecido pelo usuário poderão ser descartados.

4.5.4 Conclusões da Seção 4.5

Nesta seção foi proposto um algoritmo de caminho mínimo com restrições de tempo. Trata-se do primeiro trabalho de caminho mínimo, que se tem conhecimento, que aborda simultaneamente dois parâmetros incertos (maiores detalhes: Hernandes e Takahashi (2004)).

Ressalta-se que a $Poss(custo)$ foi calculada para efeito de ordenação, porém, caso o usuário se interesse apenas em encontrar o conjunto dos caminhos não dominados que satisfaz as restrições de tempo, então não é necessário este cálculo e, com isso, o item 6 do Passo 2 do algoritmo proposto pode ser suprimido.

Caso o usuário queira ordenar o conjunto solução através da $Poss(custo)$ este algoritmo possui uma desvantagem que necessita ser abordada: haverá casos em que o caminho que possuir menor valor modal, ou seja, $Poss(custo) = 1$ não será o que possui a maior $Poss(tempo)$, e haverá outros caminhos que são mais caros, porém que satisfarão a restrição de tempo com uma possibilidade maior. Neste caso fica a cargo do usuário decidir qual caminho escolher.

4.6 Uso da similaridade entre caminhos mínimos com incertezas

Nesta seção é implementada, no algoritmo proposto da Seção 4.4.2, a heurística de Chuang e Kung que encontra o comprimento do caminho mínimo *fuzzy* (CCMF) e o respectivo algoritmo, que determina o grau de similaridade entre o CCMF e os caminhos não-dominados.

4.6.1 Heurística de Chuang e Kung

Chuang e Kung (2005) propuseram um procedimento heurístico para encontrar o comprimento (custo) do caminho mínimo *fuzzy* (CCMF) entre todos os possíveis caminhos da rede. Este procedimento é baseado na idéia de que um número *crisp* é mínimo se e somente se todos

os outros forem maiores ou iguais a ele. No entanto, a computação associada a este procedimento, quando se trata de números *fuzzy*, é complicada. Com isso os autores propuseram um procedimento heurístico para o CCMF, reduzindo assim a complexidade computacional e também da função de pertinência. Em adição, também apresentam uma forma para encontrar qual é o caminho mínimo associado ao CCMF, baseado no cálculo do grau de similaridade entre o CCMF e cada um dos comprimentos dos caminhos existentes entre os referidos nós.

4.6.1.1 Procedimento para encontrar o CCMF (L_{min})

Assume-se que em uma rede existem m caminhos p_{rv}^t do nó r ao v , $t=1,2,\dots,m$. Se o custo é *crisp*, $l_{min}(p_{rv})$ é o mínimo entre todos os comprimentos dos caminhos $l(p_{rv}^t)$. Isto é, um número ($l(p_{rv}^t)$) é mínimo ($l_{min}(p_{rv})$) se e somente se todos os outros ($l(p_{rv}^k)$, $k \neq t$) forem maiores ou iguais a ele. Esta idéia é generalizada para o CCMF, encontrando assim o $L_{min}(p_{rv})$. Para facilitar a notação são utilizados os termos L_t e L_{min} para $L(p_{rv}^t)$ e $L_{min}(p_{rv})$, respectivamente, sendo $L(p_{rv}^t)$ o comprimento (custo) do t -ésimo caminho *fuzzy*.

Seja $\mu_{L_1}(x)$ a função de pertinência com respeito a L_1 . Analisando como número *crisp*: se x com respeito a L_1 é o menor, então não existe nenhum outro x com respeito a L_k ($k \neq 1$) menor do que ele. Esta idéia é utilizada para determinar L_{min} .

Definindo SL_1 como o conjunto *fuzzy* com respeito a L_1 e $\mu_{SL_1}(x)$ como a função de pertinência de x com respeito a SL_1 :

$$\begin{aligned} \mu_{SL_1}(x) &= \mu_{L_1}(x) \wedge \mu_{L'_2}(x) \wedge \mu_{L'_3}(x) \wedge \dots \wedge \mu_{L'_m}(x) \\ &= \min\{\mu_{L_1}(x), \mu_{L'_2}(x), \mu_{L'_3}(x), \dots, \mu_{L'_m}(x)\} \\ &= \min[\mu_{L_1}(x), \min_{k \neq 1}[\mu_{L'_k}(x)]] \end{aligned} \quad (4.2)$$

A fórmula anterior pode também ser escrita como segue:

$$SL_1 = L_1 \cap L'_2 \cap \dots \cap L'_m = L_1 \cap \left(\bigcap_{k \neq 1} L'_k \right) \quad (4.3)$$

tal que: L'_k denota o conjunto *fuzzy* de todo $y < x$ com respeito a L'_k , $k \neq 1$ (Figura 4.5), $\mu_{L'_k}$ é a função de pertinência de x com respeito a L'_k , e o operador mínimo é usado pela intersecção de dois conjuntos *fuzzy*. Além disso, $\mu_{L'_k}$ pode ser representado da forma:

$$\mu_{L'_k}(x) = 1 - \max_{y \in L_k}^{y < x} \mu_{L_1}(y) \quad (4.4)$$

Definição 22 *Seja a função de pertinência triangular de L_k representada por (m, α, β) . Dois casos poderão ocorrer para calcular L'_k :*

Caso (a): *se x estiver à direita de m (Figura 4.4).*

Para este caso, $\mu_{L'_k}(x) = 1 - \max_{y \in L_k}^{y < x} \mu_{L_k}(y) = 1 - \mu_{L_k}(m_k) = 1 - 1 = 0$

Caso (b): *se x estiver à esquerda de m .*

Para este caso, $\mu_{L'_k}(x) = 1 - \max_{y \in L_k}^{y < x} \mu_{L_k}(y) = 1 - \mu_{L_k}(x)$

*Portanto, considerando $L_k = (m_k, \alpha_k, \beta_k)$, $\mu_{L'_k}$ é chamado *half-inverse* (meia inversa), sendo definido como segue (Figura 4.5):*

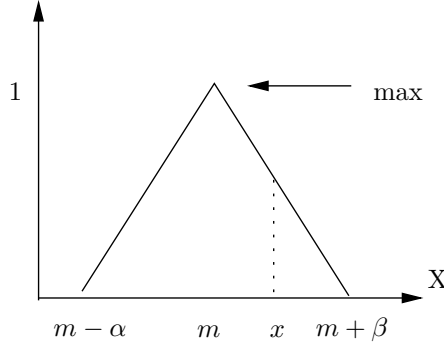


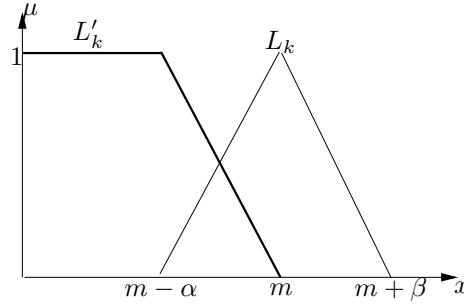
Figura 4.4: x à direita de m

$$\mu_{L'_k}(x) = \begin{cases} 0, & x > m_k \\ 1 - \mu_{L_k}(x), & m_k - \alpha_k \leq x \leq m_k \\ 1, & x < m_k - \alpha_k \end{cases} \quad (4.5)$$

$$\mu_{L_{min}}(x) = \mu_{SL_1}(x) \vee \mu_{SL_2}(x) \vee \mu_{SL_3}(x) \vee \dots \vee \mu_{SL_m}(x) = \max_{t=1}^m [\mu_{SL_t}(x)]$$

Usando a Equação 4.5 e 4.2 na equação anterior:

$$\therefore \mu_{L_{min}}(x) = \max_{t=1}^m [\min[\mu_{L_1}(x), \min_{k \neq t} \mu_{L'_k}(x)]] \quad (4.6)$$

Figura 4.5: Exemplo L'_k

O procedimento heurístico para calcular o CCMF é descrito na seção seguinte.

4.6.1.2 Procedimento heurístico para detectar o CCMF (L_{min})

Embora o procedimento descrito anteriormente (CCMF) possa ser usado para determinar o comprimento (custo) do caminho mínimo, sua computação é complicada. Devido a isso, os autores propuseram uma heurística para reduzir a complexidade computacional. Inicialmente os autores discutem o CCMF para dois caminhos, sendo o valor modal o ponto mais alto da função de pertinência (maiores detalhes: Chuang e Kung (2005)) e em seguida generalizam para m caminhos.

Durante a apresentação do algoritmo da heurística, ressalta-se que os custos dos caminhos estão denotados na seguinte forma: (valor modal, limitante inferior, limitante superior).

Algoritmo: Heurística - CCMF

PASSO 0: Dados de entrada

- $L_i = (m_i, m_i - \alpha_i, m_i + \beta_i)$, $i = 1, 2, 3, \dots, m$ (custos dos caminhos)

PASSO 1: Ordenação dos comprimentos dos caminhos

- Forme o conjunto Q ordenando os L_i 's em ordem crescente de m_i :

$$Q = \{Q^1, Q^2, Q^3, \dots, Q^m\} \text{ tal que } Q^i = (m_i, m_i - \alpha_i, m_i + \beta_i), i = 1, 2, \dots, m$$

PASSO 2: Cálculo do CCMF

1. Seja $L_{min} = (m, m - \alpha, m + \beta) = (m_1, m_1 - \alpha_1, m_1 + \beta_1) = Q^1$
2. Para $i = 2$ calcule:

- $m = \begin{cases} m, & \text{se } m \leq m_i - \alpha_i \\ \frac{(m \cdot m_i) - ((m - \alpha) \cdot (m_i - \alpha_i))}{(m + m_i) - ((m - \alpha) + (m_i - \alpha_i))}, & \text{se } m > m_i - \alpha_i \end{cases}$
- $m - \alpha = \min(m - \alpha, m_i - \alpha_i)$
- $m + \beta = \min(m + \beta, m_i)$
- $L_{min} = (m, m - \alpha, m + \beta)$

PASSO 3: $i = i + 1$

PASSO 4: Repita os Passos 2 e 3 até $i = m + 1$.

Para ilustrar a heurística anterior é apresentado um exemplo.

Exemplo 4.3 Heurística de Chuang e Kung.

Dados os custos de cinco caminhos, expressos na forma $(m, m - \alpha, m + \beta)$,

$L_1 = (262, 201, 285)$, $L_2 = (253, 196, 282)$, $L_3 = (195, 177, 256)$, $L_4 = (234, 159, 249)$ e $L_5 = (222, 160, 235)$, respectivamente, a heurística executa os seguintes passos:

PASSO 1 Ordenação dos comprimentos dos caminhos

$$Q_1 = (195, 177, 256), Q_2 = (222, 160, 235)$$

$$Q_3 = (234, 159, 249), Q_4 = (253, 196, 282)$$

$$Q_5 = (262, 201, 285)$$

PASSO 2 $L_{min} = (195, 177, 256)$

Para $i = 2$

$$\bullet m_2 = 195 > m_2 - \alpha_2 = 160 \therefore b = \frac{(195, 222) - (177, 160)}{(195 + 222) - (177 + 160)} = 187.125$$

$$\bullet m_2 - \alpha_2 = \min(177; 160) = 160; \text{ e}$$

$$\bullet m_2 + \beta_2 = \min(256; 222) = 222.$$

$$\bullet L_{min} = (187, 125, 160, 222)$$

PASSO 3 $i = i + 1$.

PASSO 4 Repita os Passos 2 e 3 até $i = 6$.

Após executada a heurística, o L_{min} encontrado foi: (179.65 160 222).

4.6.1.3 Algoritmo proposto por Chuang e Kung

Após calculado o L_{min} (Seção 4.6.1.2), Chuang e Kung (2005) usaram a intersecção entre áreas de dois triângulos para medir o grau de similaridade entre L_{min} e cada um dos L'_i s, $i = 1, 2, \dots, m$. Quanto maior a intersecção entre os dois triângulos, maior o grau de similaridade entre eles. O grau de similaridade é definido da seguinte forma:

Definição 23 O grau de similaridade entre $L_{min} = (m, \alpha, \beta)$ e $L_i = (m_i, \alpha_i, \beta_i)$ é calculado da seguinte forma:

$$s_i = \text{area}(L_i \cap L_{min}) = \begin{cases} 0, & \text{se } L_i \cap L_{min} = \emptyset \\ \frac{((m+\beta)-(m_i-\alpha_i))^2}{2[\beta+\alpha_i]}, & \text{se } L_i \cap L_{min} \neq \emptyset \end{cases}$$

Algoritmo de Chuang-Kung - 2005

PASSO 1 : Entre com uma rede direcionada com n nós e k arcos.

PASSO 2 : Encontre os caminhos possíveis entre os nós r e v com os respectivos comprimentos $L_i = (m_i, m_i - \alpha_i, m_i + \beta_i)$, $i = 1, 2, \dots, m$

PASSO 3 : Encontre o L_{min} através da heurística proposta (Seção 4.6.1.2)

PASSO 4 : Encontre o grau de similaridade s_i entre L_{min} e L_i , $i = 1, 2, \dots, m$.

PASSO 5 : Encontre o caminho com o maior grau de similaridade

PASSO 6 : FIM.

4.6.2 Adaptação do algoritmo proposto

Este algoritmo é uma adaptação do proposto na Seção 4.4.2, sendo que neste fosse incluído o algoritmo de Chuang e Kung (2005), cujo objetivo é ordenar o conjunto dos caminhos não

dominados. Tal adaptação está nos Passos 5 e 6 do referido algoritmo, por isso os demais passos não são detalhados, visto que estão na Seção 4.4.2.

Algoritmo Proposto

PASSO 1 Inicialização das variáveis.

PASSO 2 Determinação dos caminhos existentes e verificação de dominância.

PASSO 3 Critério de parada.

PASSO 4 Composição dos caminhos não dominados.

PASSO 5 Utilizando a heurística de Chuang e Kung, determinar L_{min}

PASSO 6 Encontre o grau de similaridade s_i entre L_{min} e L_i , $i = 1, 2, \dots, m$.

PASSO 7 FIM.

4.6.3 Resultados computacionais

O algoritmo proposto foi implementado na linguagem Matlab 7.0 e testado em duas redes. A primeira é a rede utilizada no trabalho de Chuang e Kung (2005) e a segunda é a rede óptica italiana de telecomunicações, testada no Exemplo 4.1.

Para continuar a notação dos exemplos anteriores, com exceção do Exemplo 4.3 que utilizou a notação $(m, m - \alpha, m + \beta)$, nos próximos exemplos a notação usada é (m, α, β) .

Exemplo 4.4 Rede de Chuang e Kung

Esta rede (Figura 4.6) foi obtida do trabalho de Chuang e Kung, cuja finalidade é comparar os resultados.

Executando o algoritmo proposto os resultados são:

1. Do nó 1 ao 2, tem-se o seguinte caminho:

- $1 \rightarrow 2$ com custo $(45 \ 12 \ 5)$

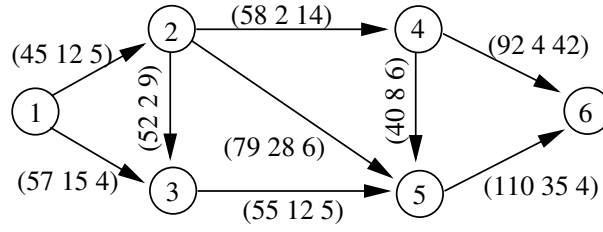


Figura 4.6: Rede de Chuang e Kung

2. Do nó 1 ao 3:

- $1 \rightarrow 3$ com custo (57 15 4)

3. Do nó 1 ao 4:

- $1 \rightarrow 2 \rightarrow 4$ com custo (103 14 19)

4. Do nó 1 ao 5:

- Caminho 1: $1 \rightarrow 3 \rightarrow 5$ com custo (112 27 9)
- Caminho 2: $1 \rightarrow 2 \rightarrow 5$ com custo (124 40 11)
- $L_{min} = (100, 72 \ 16, 72 \ 20, 28)$; $s_1 = 13, 70$ e $s_2 = 11, 35$

5. Do nó 1 ao 6:

- Caminho 1: $1 \rightarrow 2 \rightarrow 4 \rightarrow 6$ com custo (195 18 61)
- Caminho 2: $1 \rightarrow 3 \rightarrow 5 \rightarrow 6$ com custo (222 62 13)
- Caminho 3: $1 \rightarrow 2 \rightarrow 5 \rightarrow 6$ com custo (234 75 15)
- $L_{min} = (179, 65 \ 20, 65 \ 42, 35)$; $s_1 = 16, 78$, $s_2 = 18, 42$ e $s_3 = 16, 91$

Verificando os resultados anteriores e comparando-os com os resultados que constam no artigo de Chuang e Kung (2005) verifica-se que estes são parecidos, pois os caminhos que possuem grau de similaridade pequenos no artigo de Chuang e Kung são os caminhos dominados do algoritmo proposto.

Analisando os graus de similaridades entre os nós que possuem mais de um caminho não-dominado conclui-se que o caminho 1 é o melhor dentre os caminhos existentes entre os nós 1 e 5 e que o caminho 2 é o melhor dentre os caminhos dos nós 1 e 6.

Exemplo 4.5 Rede italiana

Neste exemplo a rede testada é a mesma do Exemplo 4.1, sendo que os nós estão definidos na Tabela 4.1, já os custos dos arcos estão na Tabela 4.6.

Arco	Origem	Destino	Custo
1	1	2	(115 5 22)
2	1	3	(100 6 5)
3	1	4	(140 14 27)
4	2	3	(45 10 3)
5	3	5	(40 3 9)
6	4	5	(45 6 13)
7	2	6	(40 5 5)
8	2	7	(55 10 10)
9	3	8	(47 4 3)
10	5	8	(20 5 10)
11	6	7	(20 3 4)
12	7	8	(30 5 6)
13	7	9	(40 7 6)
14	7	10	(35 4 3)
15	8	9	(20 3 7)
16	8	11	(120 10 14)
17	9	10	(15 2 4)
18	9	12	(35 5 2)

Arco	Origem	Destino	Custo
19	9	13	(63 6 3)
20	10	13	(60 7 6)
21	11	14	(50 5 5)
22	12	14	(100 8 15)
23	12	13	(25 4 3)
24	13	17	(190 17 16)
25	13	15	(100 7 7)
26	14	15	(97 6 6)
27	14	16	(100 12 9)
28	15	16	(70 7 7)
29	15	18	(35 4 6)
30	15	20	(162 9 8)
31	15	21	(165 15 18)
32	16	18	(40 3 4)
33	17	20	(160 10 10)
34	18	19	(60 5 10)
35	19	21	(70 9 4)
36	20	21	(50 6 6)

Tabela 4.6: Tabela dos arcos e custos da rede italiana (Figura 4.1)

Executando o algoritmo de Chuang e Kung para esta rede verifica-se que existem, por exemplo, 225 caminhos entre os nós 1 e 21, sendo que a maioria destes possui grau de similaridade iguais a zero. Executando o algoritmo proposto, detectou-se que entre os nós 1 e 21 existem somente dois caminhos não-dominados. Entre os demais nós existe somente um caminho não-dominado.

Portanto, analisando este exemplo para os dois algoritmos percebe-se que o de Chuang e Kung é muito custoso, pois é necessário encontrar todos os caminhos entre os nós para calcular a similaridade, enquanto que o algoritmo proposto descarta os caminhos com similaridade baixa, sem a necessidade de encontrá-los, diminuindo assim o tempo computacional.

Para ilustrar, os caminhos não dominados entre os nós 1 e 21 são:

- Caminho 1: $1 \rightarrow 3 \rightarrow 8 \rightarrow 9 \rightarrow 12 \rightarrow 13 \rightarrow 15 \rightarrow 21$ com custo (492 44 45)
- Caminho 2: $1 \rightarrow 3 \rightarrow 8 \rightarrow 9 \rightarrow 12 \rightarrow 13 \rightarrow 15 \rightarrow 18 \rightarrow 19 \rightarrow 21$ com custo (492 47 47)
- $L_{min}=(469.27 \ 24.27 \ 22.73)$; $s_1 = 14.51$ e $s_2 = 15.84$

Verificando o grau de similaridade entre os caminhos detecta-se que o segundo é o melhor, pois possui grau de similaridade maior.

4.6.4 Conclusões da Seção 4.6

O algoritmo apresenta, através da heurística e da medida de similaridade (Chuang e Kung 2005), os caminhos que podem ser considerados mínimos, ou seja, os que possuem maior grau de similaridade. Porém, ao contrário do algoritmo de Chuang e Kung, este não necessita encontrar todos os caminhos entre dois nós, visto que os caminhos que são dominados, no algoritmo proposto, apresentam grau de similaridade baixo no de Chuang e Kung. Portanto, aplicando este algoritmo em uma rede de grande porte, este deve apresentar um desempenho melhor do que o de Chuang e Kung, visto que não precisa encontrar todos os caminhos, somente os não-dominados (maiores detalhes: (Hernandes, Yamakami e Takahashi 2005)).

4.7 Algoritmo genérico para o problema de caminho mínimo com incertezas

Verificou-se que a relação de ordem do algoritmo proposto (Seção 4.4.2, relação de Okada e Soper) poderia ser substituída por outras. Com base nisto, foi adaptada uma relação de ordem genérica, ou seja, o decisor poderá escolher, ou propor, uma relação de ordem que melhor adapte ao seu problema.

Sendo as relações de ordem comparações entre dois números *fuzzy*, existem na literatura vários trabalhos envolvendo comparações entre números *fuzzy*, tanto considerando números intervalares como números *fuzzy* triangulares, que é o tipo de número *fuzzy* abordado neste trabalho. Problemas de otimização com número intervalar e restrições *fuzzy* são abordados em trabalhos como os de Delgado, Verdegay e Vila (1988) e de Ishibuchi e Tanaka (1990). Já com relação aos números *fuzzy*, como por exemplo números triangulares, existem os trabalhos de Dubois e Prade (1983), Cheng (1998), Delgado et al. (1988) e Okada e Soper (2000).

4.7.1 Relação de ordem

Como citado, o algoritmo proposto possui uma relação de ordem genérica, sendo definida da seguinte forma:

Definição 24 *Dados dois números fuzzy triangulares $\tilde{a}$ e $\tilde{b}$, então $\tilde{a} \prec \tilde{b}$ ($\tilde{a}$ domina $\tilde{b}$) se e somente se $f(\tilde{a}) < f(\tilde{b})$.*

A função $f(*)$ é a comparação entre números *fuzzy*, podendo ser aplicada tanto em relações de ordem que trabalham com os números *fuzzy*, quanto em relações que utilizam índices de defuzzificação.

O algoritmo proposto foi implementado utilizando as seguintes relações de ordem: primeiro índice de Yager (Yager 1978, Yager 1980, Yager 1981), índice de Liou e Wang (1992), índice de García e Lamata (2005), relação de ordem de Okada e Soper (2000), índice de Nayeem e Pal (2005) e o índice de possibilidade de Dubois e Prade (1983), relações estas descritas na Seção 2.7 do Capítulo 2.

4.7.2 Algoritmo com relação de ordem genérica

A adaptação da relação genérica ocorre no Item 3 do Passo 2 do algoritmo da Seção 4.4.2.

Informações sobre o algoritmo proposto

V : conjunto de nós;

it : contador de iterações;

$(m + \beta)^i$: espalhamento à direita $(m + \beta)$ do custo do nó i ;

$\tilde{l}_{ji}$: custo do arco (j, i) ;

$\tilde{c}_{(i,k)}^{it}$: custo do caminho entre os nós 1 e i com a etiqueta k , na iteração it ;

M : um número com valor grande, substitui o ∞ do algoritmo clássico de Ford-Moore-Bellman;

Γ_i^{-1} : conjunto dos nós predecessores de i ; e

$f(\tilde{c}_{i,k}^{it})$: relação de ordem aplicada ao comprimento do caminho entre os nós 1 e i , com etiqueta k , na iteração it .

Algoritmo

PASSO 1: Inicialização das variáveis

1. $\tilde{c}_{(1,1)}^0 = (0, 0, 0)$
2. $\tilde{c}_{(j,1)}^0 = (M + 2, 1, 1)$, $j = 2, 3, \dots, r$

tal que:

- r : números de nós;
- $M = \sum_{i=1}^{na} |(m + \beta)^i|$; e
- na : número de arcos.

3. $it \leftarrow 1$.

PASSO 2: Determinação dos caminhos e verificação da dominância

1. $\tilde{c}_{(1,1)}^{it} = (0, 0, 0)$
2. $\forall j \in \Gamma_i^{-1}$, $i = 2, 3, \dots, r$, faça:

- $\tilde{c}_{(i,k1)}^{it} = \tilde{c}_{(j,k2)}^{it-1} \oplus \tilde{l}_{ji}$;

3. Verificação da dominância entre as etiquetas do nó i :

Para todas as etiquetas do nó i faça:

- Se $f(\tilde{c}_{(i,k1)}^{it}) > f(\tilde{c}_{(i,k2)}^{it}) \Rightarrow$ elimine a $k1$ -ésima etiqueta
- Se $f(\tilde{c}_{(i,k1)}^{it}) < f(\tilde{c}_{(i,k2)}^{it}) \Rightarrow$ elimine a $k2$ -ésima etiqueta

PASSO 3: Critério de parada

1. Se $(\tilde{c}_{(i,k1)}^{it} = \tilde{c}_{(i,k1)}^{it-1} \forall i \in V)$ ou $(it = r)$ faça:

- Se $it = r$ e $\tilde{c}_{(i,k1)}^{it} \neq \tilde{c}_{(i,k1)}^{it-1} \Rightarrow$ Passo 5 (circuito negativo)
- Senão $\Rightarrow$ Passo 4

2. Senão: $it \leftarrow it + 1 \Rightarrow$ volte ao Passo 2

PASSO 4: Composição dos caminhos

- Encontre todos os caminhos não-dominados entre os nós 1 e i

PASSO 5: FIM.

4.7.3 Resultados computacionais

O algoritmo proposto foi implementado no Matlab 7.0 e, para ilustrar, foi executado em duas redes, usando as relações de ordem comentadas anteriormente para os seguintes valores: primeiro índice de Yager. Índice de Liou e Wang para $\lambda = 0, 0.5$ e 1 . Índice García e Lamata para $\lambda = 0$ e $\delta = 0$; $\lambda = 0$ e $\delta = 0.5$; $\lambda = 0$ e $\delta = 1$; $\lambda = 0.5$ e $\delta = 0$; $\lambda = 0.5$ e $\delta = 0.5$; $\lambda = 0.5$ e $\delta = 1$; $\lambda = 1$ e $\delta = 0$; $\lambda = 1$ e $\delta = 0.5$; e $\lambda = 1$ e $\delta = 1$. Okada e Soper para $\epsilon = 0$ e 0.5 . Nayeem e Pal e possibilidade de Dubois e Prade.

Exemplo 4.6 Rede com custos positivos

Dada a rede da Figura 4.7 e seus respectivos custos (Tabela 4.7), a Tabela 4.8 apresenta os resultados.

Nó origem	Nó destino	Custo do arco
1	2	(820 20 20)
1	3	(361 11 9)
1	6	(677 27 6)
1	9	(300 10 50)
1	10	(450 30 20)
2	3	(186 6 7)
2	5	(510 15 15)
2	9	(930 30 30)
3	4	(667 17 216)
3	5	(748 18 22)
3	8	(443 18 22)
4	5	(199 9 11)
4	6	(340 30 20)

Nó origem	Nó destino	Custo do arco
4	11	(740 30 30)
5	6	(660 50 30)
6	11	(242 12 18)
7	6	(410 20 30)
7	11	(472 22 18)
8	4	(730 20 5)
8	7	(242 12 13)
9	8	(137 7 8)
9	7	(130 10 20)
9	10	(242 12 18)
10	7	(342 12 8)
10	11	(1310 60 130)

Tabela 4.7: Custos dos arcos do Exemplo 4.6

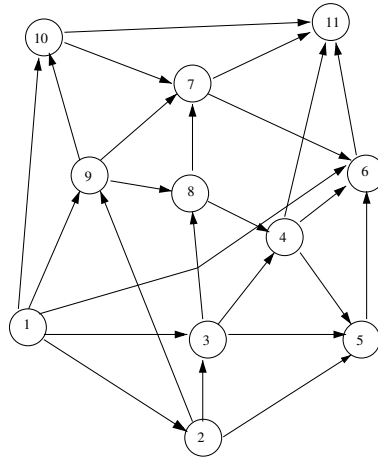


Figura 4.7: Rede exemplo

Nó destino	Caminho mínimo	Custo do Caminho	Relação de ordem
2	1→ 2	(820 20 20)	Todas
3	1→ 3	(361 11 9)	Todas
4	1→ 3→4	(1028 28 225)	Todas
4	1→ 9→ 8→4	(1167 37 63)	Okada e Soper ($\epsilon = 0$)
5	1→ 3→5	(1109 29 31)	Todas
6	1→ 6	(677 27 6)	Todas
7	1→ 9→7	(430 20 70)	Todas
8	1→ 9→8	(437 17 58)	Todas
9	1→ 9	(300 10 50)	Todas
10	1→ 10	(450 30 20)	Todas
11	1→ 9→ 7→11	(902 42 88)	Yager; Liou e Wang ($\lambda = 1$); García e Lamata($\lambda = 1, \delta = 0$); e Okada e Soper($\epsilon = 0$ e 0.5).
11	1→ 6→11	(919 39 24)	Liou e Wang ($\lambda = 0$ e 0.5); García e Lamata (exceto $\lambda = 1, \delta = 0$); Okada e Soper($\epsilon = 0$ e 0.5); Nayeem e Pal; e Dubois e Prade.

Tabela 4.8: Resultados da rede com custos positivos (Exemplo 4.6).

Comentários:

Nota-se que todos os índices de ordenação produzem os mesmos caminhos entre os nós 1 e i ($i = 2, 3, \dots, 10$), com a exceção da relação de Okada e Soper que apresenta um caminho adicional entre os nós 1 e 4. Porém, entre os nós 1 e 11 os resultados mostram que há dois diferentes caminhos, sendo um com custo (919, 39, 24) outro com custo (902, 42, 88) .

Exemplo 4.7 Rede com custo negativo

Este algoritmo foi executado para a rede da Figura 4.8 e apresentou os resultados da Tabela 4.9.

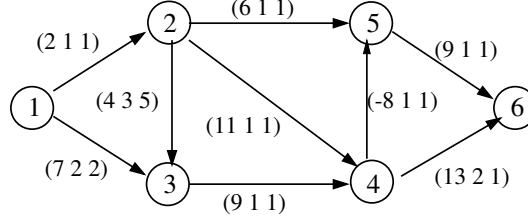


Figura 4.8: Rede negativa

Nó destino	Caminho mínimo	Custo do caminho	Relação de ordem
2	1 → 2	(2 1 1)	Todas
3	1 → 3	(7 2 2)	Okada e Soper ($\epsilon = 0$ e 0.5); Liou e Wang ($\lambda = 1$); e García e Lamata ($\lambda = 1, \delta = 0$ e $\lambda = 1, \delta = 0.5$).
3	1 → 2 → 3	(6 4 6)	Todas, exceto: Liou e Wang ($\lambda = 1$); e García e Lamata ($\lambda = 1, \delta = 0$).
4	1 → 2 → 4	(13 2 2)	Todas
4	1 → 2 → 3 → 4	(15 5 7)	Okada e Soper ($\epsilon = 0$)
5	1 → 2 → 4 → 5	(5 3 3)	Todas
5	1 → 2 → 3 → 4 → 5	(7 6 8)	Okada e Soper ($\epsilon = 0$)
6	1 → 2 → 4 → 5 → 6	(14 4 4)	Todas
6	1 → 2 → 3 → 4 → 5 → 6	(16 7 9)	Okada e Soper ($\epsilon = 0$)

Tabela 4.9: Resultados da rede negativa - Exemplo 4.7

Comentários:

Analisando os resultados da Tabela 4.9, observa-se que algumas relações de ordem apresentam os mesmos resultados. Portanto, são comentadas as relações que possuem diferenças, donde destaca-se:

a) Inicialmente verifica-se que a relação de Okada e Soper, para ϵ igual a zero, apresenta um número considerável de caminhos não-dominados, caso que não acontece para ϵ igual 0.5. Logo, conclui-se que há dominância parcial com alguns caminhos.

b) Verifica-se que o índice de Liou e Wang é um caso particular do índice de García e Lamata (quando $\delta = 0$). Também percebe-se que o índice de García e Lamata para $\lambda = 1$

e $\delta = 0.5$ apresenta dois caminhos diferentes entre os nós 1 e 3. A justificativa é que para estes valores de λ e δ é somente verificado o espalhamento da esquerda dos caminhos ($1 \rightarrow 3$, $1 \rightarrow 2 \rightarrow 3$), sendo assim, estes apresentam o mesmo índice.

4.7.4 Conclusões da Seção 4.7

Nesta seção foi apresentada uma adaptação para o algoritmo da Seção 4.4.2, onde é apresentada uma relação de ordem genérica, ficando a cargo do usuário implementar a que melhor adaptar ao seu problema (maiores detalhes: Hernandes, Lamata, Takahashi, Verdegay e Yamakami (2006), Hernandes, Lamata, Verdegay e Yamakami (2006)).

4.8 Considerações finais

Foi proposto neste capítulo um algoritmo para o problema de caminho mínimo em grafos com incertezas nos custos dos arcos, sendo que este possui algumas vantagens sobre os da literatura: pode ser executado em uma rede com custos quaisquer; detecta a existência de circuito negativo; possui complexidade computacional menor do que os da literatura.

Além disso, o algoritmo pode ser implementado utilizando a relação de ordem que o usuário desejar, podendo trabalhar com números *crisp* usando índices de defuzzificação, ou mesmo com números *fuzzy*, visto que possui uma relação de ordem genérica. Dependendo da relação de ordem adotada, o algoritmo poderá encontrar tanto um conjunto solução de caminhos quanto um único.

O algoritmo proposto também possui a vantagem de poder ser aplicado em redes que possuem dois parâmetros incertos, como ocorreu com a restrição de tempo.

Capítulo 5

O Problema de Fluxo Máximo com Incertezas

5.1 Introdução

Aplicações em áreas de telecomunicações; transportes; distribuição de energia; computação; manufatura; dentre outras (Gondran e Minoux 1984, Goldbarg e Luna 2000), fazem com que os problemas de fluxo em redes sejam bastante estudados na área da teoria dos grafos.

Dentre os problemas de fluxo em redes destaca-se o problema de fluxo máximo, que consiste em enviar a maior quantidade de fluxo entre dois nós (origem s e destino t) de uma rede, respeitando as restrições de capacidade de cada arco. Estes problemas possuem aplicações em muitas áreas, tais como: distribuição (por exemplo: maximizar o fluxo de uma rede de distribuição de uma companhia a partir de suas fábricas até os seus consumidores; maximizar a distribuição de água de um sistema de aquedutos, etc), transportes (por exemplo: maximizar o fluxo de veículos de uma rede de transporte), dentre outras.

Em se tratando deste problema em uma rede clássica, ou seja, em uma rede com parâmetros *crisp*, existem, na literatura, algoritmos eficientes para resolvê-lo (Ahuja et al. 1993, Bazarra et al. 1990, Goldbarg e Luna 2000). Porém, há problemas em que alguns parâmetros da rede (custos, capacidades, demandas, etc.) são incertos, transformando assim o problema em um chamado *problema de fluxo máximo fuzzy*.

O número de trabalhos da literatura que abordam o problema de fluxo máximo em redes com estrutura *crisp* e as capacidades dos arcos incertas é reduzido, devendo-se a Kim e Roush

(1982), Chanas e Kolodziejczyk (Chanas e Kolodziejczyk 1982, Chanas e Kolodziejczyk 1984, Chanas e Kolodziejczyk 1986) e Takahashi (2004). Figurando entre os primeiros trabalhos da área, 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*. Já Chanas e Kolodziejczyk (Chanas e Kolodziejczyk 1982, Chanas e Kolodziejczyk 1984, Chanas e Kolodziejczyk 1986), que são os principais trabalhos envolvendo este tema, abordam este problema utilizando a técnica dos cortes mínimos, ou seja, encontram os cortes essenciais da rede e determinam o politopo formado pela intersecção de todas as capacidades dos cortes essenciais. Após encontrado o politopo, determina-se o fluxo máximo, que é o maior ponto da intersecção entre o politopo encontrado e o fluxo máximo desejável, fluxo este fornecido pelo usuário. O primeiro trabalho de Chanas e Kolodziejczyk (1982) foi desenvolvido para um grafo com estrutura *crisp* e arcos com capacidades incertas, isto é, os arcos têm uma função de pertinência associada ao fluxo, dependente da capacidade. No segundo trabalho, Chanas e Kolodziejczyk (1984) abordam este problema para o caso do fluxo ser um número real *fuzzy*, sendo que a capacidade inferior e a superior têm pertinências associadas. A teoria para o fluxo inteiro *fuzzy* foi desenvolvida em Chanas e Kolodziejczyk (1986). Figurando entre os mais recentes trabalhos, Takahashi (2004) propôs dois algoritmos para este tipo de problema, o primeiro é semelhante aos propostos por Chanas e Kolodziejczyk, onde é encontrado o politopo formado pela intersecção das capacidades dos cortes essenciais do grafo. Já no segundo algoritmo é proposta uma heurística, onde são aplicados α -cortes nas capacidades, transformando o problema em um problema clássico e em seguida utiliza-se um algoritmo clássico de corte mínimo.

Analisando os trabalhos citados anteriormente, foi verificado que estes apresentam algumas desvantagens, tais como: apresentam como solução final somente o fluxo máximo e os nós que pertencem a cada corte essencial, não apresentando o comportamento do fluxo ao longo dos arcos da rede; exigem que o usuário forneça o fluxo máximo desejável e sua respectiva função de pertinência (é discutido adiante, Seção 5.2.2) e a partir daí verificam o grau de satisfação deste fluxo na rede; e, não se pode garantir a eficiência quando aplicados em

redes de grande porte.

Com base nas desvantagens supracitadas, apresenta-se neste trabalho um algoritmo para o problema de fluxo máximo *fuzzy* que utiliza a técnica dos caminhos aumentantes e não possui as desvantagens comentadas anteriormente, visto que apresenta na solução final o fluxo máximo e o comportamento do mesmo ao longo dos arcos da rede, além de não exigir que o usuário apresente o fluxo máximo desejável, o que é muito importante trata-se de uma rede de grande porte.

O algoritmo proposto é uma adaptação do algoritmo clássico de Ford e Fulkerson (Ahuja et al. 1993, Goldberg e Luna 2000), sendo que as incertezas dos parâmetros (capacidades) são abordadas utilizando a teoria dos conjuntos *fuzzy*.

Com relação ao problema de fluxo máximo em redes com estruturas incertas *fuzzy*, que não é abordado neste trabalho, o único trabalho existente na literatura é o de Chanas et al. (1995), onde os autores abordam este problema em um de transporte *fuzzy*. Neste trabalho estão reunidos os trabalhos de Chanas e Kolodziejczyk (Chanas e Kolodziejczyk 1982, Chanas e Kolodziejczyk 1984, Chanas e Kolodziejczyk 1986) e Delgado, Verdegay e Vila (1985) (maiores detalhes: Malik e Moderson (2001)), onde são definidos os problemas de fluxo máximo *fuzzy*, de rede *fuzzy* e de transporte *fuzzy*.

Este capítulo está organizado da seguinte forma: na Seção 2 é formulado o problema de fluxo máximo, tanto considerando o problema clássico quanto o problema com incertezas. O algoritmo proposto e o algoritmo de Chanas e Kolodziejczyk (1984), que é o principal algoritmo para este problema que trata o fluxo como um número real, estão na Seção 3. Na Seção 4 são apresentados alguns resultados computacionais. Conclusões são discutidas na Seção 5.

5.2 O problema de fluxo máximo

Nesta seção são abordados os problemas de fluxo máximo, tanto considerando o problema clássico quanto em uma rede *crisp* com capacidades incertas.

5.2.1 O problema de fluxo máximo clássico

Dada uma rede direcionada $G = (V, A)$ com arcos capacitados superiormente (denotados por u_{ij}) e associando a cada arco (i, j) um valor x_{ij} , correspondente à quantidade de fluxo percorrido no arco (i, j) , o problema de fluxo máximo tem que satisfazer as seguintes restrições:

1. $x_{ij} \geq 0$ (fluxo não negativo em cada arco);
2. $x_{ij} \leq u_{ij}$ (restrições de capacidade); e
3. $\sum_i x_{ij} = \sum_k x_{jk}, \forall j \neq s, t$ (satisfaz a regra de conservação do fluxo em cada nó, isto é, o fluxo que entra em cada nó é igual ao que deixa).

Logo, este problema pode ser formulado da seguinte forma:

$$\begin{aligned} & \text{Max} && v \\ & \text{s.a :} && \sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(i,j) \in A} x_{jk} = \begin{cases} v, & j = s \\ 0, & \forall j \in V - \{s, t\} \\ -v, & j = t \end{cases} \\ & && 0 \leq x_{ij} \leq u_{ij} \quad (i, j) \in A \end{aligned}$$

tal que:

v : quantidade máxima de fluxo que pode percorrer a rede do nó origem s ao nó destino t .

Dado um caminho qualquer de s a t numa rede G ($p = \{s = i_1(i_1, i_2)i_2, \dots, i_{k-1}, (i_{k-1}, i_k)i_k = t\}$), a quantidade máxima de fluxo que pode ser enviada de s a t pelo caminho p , satisfazendo as restrições de capacidade é:

$$\min\{u_{ij} \mid (i, j) \in p\}.$$

O arco de p com a menor capacidade fica saturado, não podendo passar mais fluxo por ele.

Este tipo de problema pode ser resolvido utilizando duas técnicas distintas, a técnica dos *cortes mínimos* e a dos *caminhos aumentantes*. A primeira consiste em dividir o conjunto dos nós em dois subconjuntos, V_1 e V_2 , tal que o nó origem s pertença a V_1 e o nó destino t a V_2 . Esta técnica encontra dentre todos os possíveis cortes do grafo aquele que possui a menor capacidade, sendo a capacidade do corte a soma das capacidades dos arcos que possuem o

nó origem em V_1 e o destino em V_2 . Logo, a capacidade deste corte será o fluxo máximo que pode ser enviado de s a t . Já a técnica dos caminhos aumentantes determina um caminho existente entre os nós s e t , analisa as capacidades de cada arco deste caminho, caso estes não estejam saturados se adiciona fluxo até saturá-los. Este processo se repete até quando não for mais possível encontrar um caminho entre s e t .

5.2.2 O problema de fluxo máximo *fuzzy*

Na maioria das aplicações deste problema, as capacidades dos arcos, que são números trapezoidais *fuzzy*, são somente limitadas superiormente, o que também ocorre neste trabalho. Logo os limitantes inferiores ($m_1 - \delta$) e extremos inferiores do intervalo modal (m_1) são iguais a zero para todas as capacidades, conforme exemplifica a Figura 5.1.

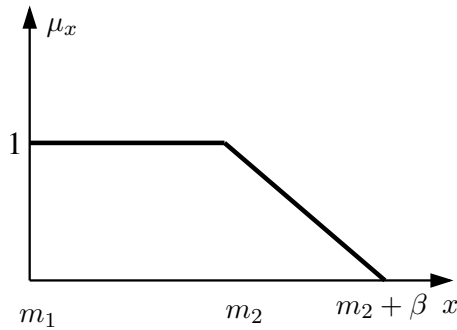


Figura 5.1: Exemplo de capacidade *fuzzy*

Analisando a Figura 5.1, pode-se dizer que se o fluxo que percorrer o arco (i, j) for menor ou igual a m_2 , seu grau de satisfação é 1, ou seja, estará satisfazendo a restrição de capacidade daquele arco. Caso o fluxo esteja entre m_2 e $m_2 + \beta$, o seu índice de satisfação estará entre 1 e zero, isto é, estará satisfazendo parcialmente a restrição de capacidade daquele arco. Com base nisto, se implementou uma heurística, baseada em α -cortes (Takahashi 2004), que transforma este problema *fuzzy* em um problema de fluxo máximo clássico, onde o usuário poderá escolher qual é o grau de satisfação mínimo que deseja que suas restrições de capacidade sejam satisfeitas.

Assim sendo, a cada restrição de capacidade $x_{ij}^v \leq u_{ij}$ ($x_{ij}^v = \{x_{ij} : (i, j) \in A, x_{ij} \geq 0\}$) pode ser associada uma função de pertinência:

$$\mu_{ij}(x_v) = \begin{cases} 1 & \text{se } x_{ij}^v < u_{ij} \\ \bar{L}(u_{ij}, \bar{u}_{ij}; x_{ij}^v) & \text{se } u_{ij} \leq x_{ij}^v \leq \bar{u}_{ij} \\ 0 & \text{se } x_{ij}^v > \bar{u}_{ij} \end{cases} \quad (5.1)$$

tal que $\mu_{ij}(x_v)$ fornece o *grau de confiança* da satisfação da restrição $x_{ij}^v \leq u_{ij}$.

Como citado na introdução, os principais trabalhos da literatura exigem que o usuário forneça o fluxo ideal, logo é associada uma função de satisfação ao atendimento deste fluxo (Figura 5.2), como sendo:

$$\mu_S(v) = \begin{cases} 1 & \text{se } v > v_0 \\ L(v_1, v_0; v) & \text{se } v_1 \leq v \leq v_0 \\ 0 & \text{se } v < v_1 \end{cases} \quad (5.2)$$

tal que: v_0 é a quantidade mínima ideal de fluxo exigido pelo usuário, v_1 é a quantidade mínima aceitável de fluxo e $L(v_0, v_1; v)$ é uma função linear, sendo $L(v_0, v_1; v_0) = 1$ 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 .

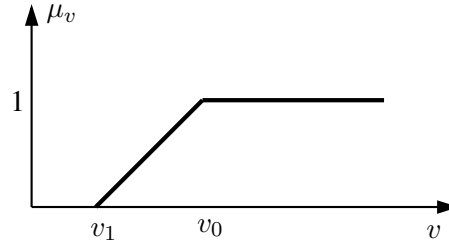


Figura 5.2: Exemplo de uma função que representa $\mu_S(v)$

Logo, o problema de fluxo máximo *fuzzy*, segundo Chanas e Kolodziejczyk, pode ser reduzido a encontrar um fluxo que maximize o valor da função de pertinência $\mu_D(x_v)$ em uma decisão da forma (Bellman e Zadeh 1970):

$$\mu_D(x_v) = \mu_C(x_v) \bigwedge \mu_S(x_v) \quad (5.3)$$

$$\mu_C(x_v) = \bigwedge_{(i,j) \in A} \mu_{ij}(x_{ij}) \quad (5.4)$$

tal que: $\mu_C(x_v)$ denota o *grau de satisfação simultânea das restrições e capacidade em todos os arcos da rede pelo fluxo* x_v .

Esta definição (Figura 5.3) já foi discutida no Capítulo 3.

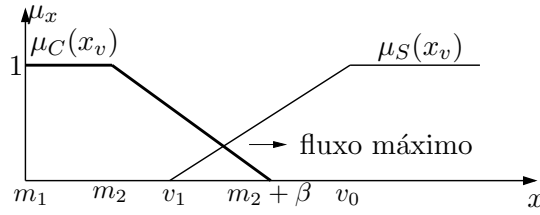


Figura 5.3: Exemplo de função de satisfação

Definição 25 Um fluxo x_v^* é chamado de *fluxo máximo* se:

$$x_v^* = \max_{x_v} \mu_D(x_v) \quad (5.5)$$

Definição 26 O fluxo x_v com o maior v dentre os fluxos máximos é chamado de *fluxo ótimo*.

5.3 Algoritmos para o problema de fluxo máximo com incertezas

Esta seção apresenta o principal algoritmo da literatura, proposto por Chanas e Kolodziejczyk (1984), e o algoritmo proposto neste trabalho.

5.3.1 Algoritmo proposto por Chanas e Kolodziejczyk (1984)

O algoritmo de Chanas e Kolodziejczyk foi desenvolvido para o problema de fluxo contínuo em redes com capacidade *fuzzy*, sendo estas, como citado, números trapezoidais. Este algoritmo possui como objetivo verificar com que grau de satisfação o fluxo máximo desejável, fornecido pelo usuário (ou seja, o fluxo que o usuário crê ser ideal), pode ser percorrido na rede.

Visto que o problema em questão possui incertezas nas capacidades, então um intervalo de fluxos admissíveis no arco $(i, j) \in A$ não é dado pelo intervalo $[b_{ij}, c_{ij}]$, mas sim na forma de um intervalo *fuzzy*, denotado por C_{ij} . Já 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} do arco (i, j) .

Definição 27 Um conjunto fuzzy V sobre $\mathbb{R}$ com a função de pertinência $\mu_V(v) = \bigvee_{x_v} \mu_C(x_v)$ é chamada *capacidade fuzzy da rede*.

Definição 28 A capacidade *emphfuzzy* do corte (V_1, V_2) , para uma rede com capacidade superior, é o número $C(V_1, V_2)$ definido como:

$$C(V_1, V_2) = \sum_{(i,j) \in \overline{(V_1, V_2)}} C_{ij} \quad (5.6)$$

Teorema 5.1 Seja W o conjunto de todos os cortes da rede e I o conjunto fuzzy tal que $\mu_I(x) = \bigvee_v \mu_V(v)$, $\forall x \in \mathbb{R}$. Então a seguinte relação vale:

$$V = I \cap \left(\bigcap_{(V_1, V_2) \in W} C(V_1, V_2) \right) \quad (5.7)$$

Prova: Consultar Chanas e Kolodziejczyk (1984) ■

As definições comentadas anteriormente dão base para o seguinte algoritmo:

Informações sobre o algoritmo:

x_w : fluxo máximo fornecido pelo usuário; e

$\mu_R(x_w)$: pertinência do fluxo x_w .

Algoritmo de Chanas e Kolodziejczyk

PASSO 1 Determinar o fluxo real máximo x_w e o respectivo corte mínimo (V_1, V_2) na rede com capacidade u_{ij} . Se $\mu_R(x_w) = 1$, então x_w é ótimo. Caso contrário, vá ao Passo 2.

PASSO 2 Determinar o número *fuzzy* $C(V_1, V_2)$ e as coordenadas (v, p) do ponto de intersecção das funções $\mu_{C(V_1, V_2)}$ e μ_S . Se $p=0$, então o problema é sem solução, FIM. Caso contrário vá ao Passo 3.

PASSO 3 Determinar o fluxo máximo x_w e o respectivo corte mínimo (V_1, V_2) 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 volte ao Passo 2.

5.3.2 Algoritmo Proposto

O algoritmo proposto utiliza a técnica dos caminhos aumentantes e aborda as incertezas existentes nas capacidades aplicando α -cortes, baseado na heurística proposta por Takahashi

(2004). Este algoritmo não exige que o usuário forneça o fluxo máximo desejável x_w , o que é uma grande vantagem quando se trata de uma rede de grande porte.

O algoritmo é composto dos seguintes passos: o primeiro é a atribuição do grau de pertinência mínimo ($\bar{\alpha}$) desejado pelo usuário para o fluxo máximo e o número de partições de α -cortes. No Passo 2 é aplicada a heurística baseada em α -cortes e atribuído o fluxo inicial factível. É atribuído fluxo nulo como o inicial factível. O Passo 3 é destinado à aplicação de um algoritmo clássico para encontrar um caminho entre o nó origem s e o destino t . No Passo 4 é verificada a existência de alguma cadeia aumentante entre os nós origem e destino, caso exista, determina-se o fluxo aumentante e o inclui nos arcos da cadeia. O algoritmo é finalizado no Passo 5.

Informações sobre o algoritmo:

V : conjunto dos vértices (nós);

$\bar{\alpha}$: grau de pertinência mínimo que o usuário deseja para o fluxo máximo;

H : número de partições do intervalo $[\bar{\alpha}, 1]$;

h : tamanho do subintervalo $[\alpha_k, \alpha_{k+1}]$;

$(c_{ij})_\alpha$: valor *crisp* da capacidade do arco (i, j) com pertinência α ;

(t, s) : arco artificial, que une o nó destino t ao nó origem s ;

$(x_{ij})_\alpha$: quantidade de fluxo que percorre o arco (i, j) com pertinência α ;

T : circuito formado pela união do caminho encontrado entre s e t com o arco artificial (t, s) ;

Γ : ciclo formado entre a cadeia que une os nós s e t , juntamente com o arco artificial (t, s) ;

Γ^+ : arcos de Γ orientados no sentido do arco (t, s) ; e

Γ^- : arcos de Γ orientados no sentido contrário do arco (t, s) .

Algoritmo Proposto

PASSO 1 Inicialização das variáveis (atribuição do grau de pertinência mínimo desejado pelo usuário e o número de partições de α -cortes)

- Fornecer $\bar{\alpha}$ e H

PASSO 2 Defuzzificação das capacidades e atribuições dos fluxos iniciais factíveis

- Para cada $\alpha = \bar{\alpha} + i * h$ ($h = \frac{1-\bar{\alpha}}{H-1}$, $i = 0, 1, \dots, H-1$) encontrar $(c_{ij})_\alpha$;
e
- $(x_{ij})_\alpha \rightarrow 0$, $\forall (i, j) \in V$ (atribuição do fluxo inicial factível).

PASSO 3 Determinação de um caminho entre s e t

Repetir este passo até quando não for possível encontrar um caminho entre s e t .

- Obter um caminho p de s a t ao longo do qual nenhum arco esteja saturado e fazer:
 - (a) Encontrar o circuito T formado pela união entre p e (t, s)
 - (b) Aumentar o fluxo ao longo do circuito em

$$\Delta = \min_{(i,j) \in T} [(c_{ij})_\alpha - (x_{ij})_\alpha]$$
 - (c) Eliminar o(s) arco(s) saturado(s) do caminho

PASSO 4 Determinação de uma cadeia aumentante

Se existir uma cadeia entre s e t fazer:

- (a) Determinar $\delta = \min\{\delta_1, \delta_2\}$ tal que:

- $\delta_1 = \min_{(i,j) \in \Gamma^+} [(c_{ij})_\alpha - (x_{ij})_\alpha]$
- $\delta_2 = \min_{(i,j) \in \Gamma^-} [(x_{ij})_\alpha]$

- (b) Se $\delta > 0$, fazer:

- $\forall (i, j) \in \Gamma^+ \Rightarrow (x_{ij})_\alpha \rightarrow (x_{ij})_\alpha + \delta$
- $\forall (i, j) \in \Gamma^- \Rightarrow (x_{ij})_\alpha \rightarrow (x_{ij})_\alpha - \delta$

PASSO 5 FIM.

Como o algoritmo proposto trabalha com α -cortes, então a função satisfação ao atendimento de um certo fluxo, $\mu_s(x_v)$ (Seção 5.2.2) é o valor de α do algoritmo proposto.

5.4 Resultados computacionais

O algoritmo proposto foi executado na rede contida em Chanas e Kolodziejczyk (1984), visto que os resultados dos dois algoritmos são comparados (maiores detalhes, vide Hernandez, Yamakami, Takahashi e Verdegay (2006)).

Ressalta-se que os valores das capacidades dos arcos (Figura 5.4) estão expressos na forma $(m_1 - \delta, m_1, m_2, m_2 + \beta)$, sendo que $m_1 - \delta = m_1 = 0$.

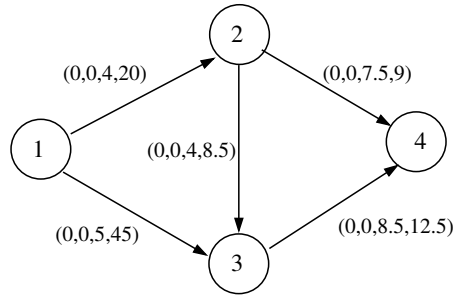


Figura 5.4: Rede CK(84)

Esta rede foi utilizada em ambos os algoritmos, obtendo os seguintes resultados:

a) Resultados do algoritmo de Chanas e Kolodziejczyk (1984)

Considerando o fluxo máximo desejável $G[v_1, v_0] = [18, 20]$ (Equação 5.2), fornecido pelo usuário, e conseqüentemente $\mu_S(v)$ da forma:

$$\mu_S(v) = \begin{cases} 1, & \text{se } v > 20 \\ L(18, 20; v), & \text{se } 18 \leq v \leq 20, \\ 0, & \text{se } v < 18 \end{cases}$$

o fluxo máximo obtido entre os nós 1 e 4 foi $v = 18, 93$, com grau de pertinência $\mu_D(18, 93) = 0,467$ (ver Figura 5.6).

b) Resultados do algoritmo proposto

Executando a mesma rede (Figura 5.4) no algoritmo proposto, para $H = 10$ e $\bar{\alpha} = 0$, obtém-se os resultados da Tabela 5.1.

H=10 e $\bar{\alpha} = 0$	Arco (1,2)	Arco (1,3)	Arco (2,3)	Arco (2,4)	Arco (3,4)	Fluxo Máximo
$\alpha = 0$	17,5	4	8,5	9	12,5	21,5
$\alpha = 0,1$	16,9	4,05	8,05	8,85	12,1	20,95
$\alpha = 0,2$	16,3	4,1	7,6	8,7	11,7	20,4
$\alpha = 0,3$	15,2	4,65	6,65	8,55	11,3	19,85
$\alpha = 0,4$	13,6	5,7	5,2	8,4	10,9	19,3
$\alpha = 0,5$	12	6,75	3,75	8,25	10,5	18,75
$\alpha = 0,6$	10,4	7,8	2,3	8,1	10,1	18,2
$\alpha = 0,7$	8,8	8,85	0,85	7,95	9,7	17,65
$\alpha = 0,8$	7,2	9,3	0	7,2	9,3	16,5
$\alpha = 0,9$	5,6	8,9	0	5,6	8,9	14,5
$\alpha = 1$	4	5	0	4	5	9

Tabela 5.1: Resultados do algoritmo proposto para $\bar{\alpha} = 0$ e $H = 10$

H=2 e $\bar{\alpha} = 0,5$	Arco (1,2)	Arco (1,3)	Arco (2,3)	Arco (2,4)	Arco (3,4)	Fluxo Máximo
$\alpha = 0,5$	12	6,75	3,75	8,25	10,5	18,75
$\alpha = 0,75$	8	9,375	0,125	7,875	9,5	17,375
$\alpha = 1$	4	5	0	4	5	9

Tabela 5.2: Resultados do algoritmo proposto para $\bar{\alpha} = 0,5$ e $H = 2$

Executando o algoritmo para $H = 2$ e $\bar{\alpha} = 0,5$, obtém-se os seguintes resultados da Tabela 5.2.

c) Comparação dos resultados

Verificando os resultados dos dois algoritmos percebe-se que o resultado de Chanas e Kolodziejczyk está contido nos resultados do algoritmo proposto, pois este foi verificado para vários α -cortes, enquanto que o de Chanas e Kolodziejczyk foi testado para somente um $\mu_S(x_v)$ fornecido pelo usuário.

Através dos resultados da Tabela 5.1 também foi possível verificar como é o comportamento dos arcos à medida em que o valor de α aumenta. Percebe-se, por exemplo, que o arco (2,3) é um arco auxiliar, pois à medida que α aumenta, a quantidade de fluxo que passa por ele diminui. Efetuando esta mesma análise nos resultados de Chanas e Kolodziejczyk verifica-se que o corte essencial cujo arco (2,3) pertence é exatamente o corte que não influencia na construção do politopo solução.

Analisando as Figuras 5.5 e 5.6, que esboçam as soluções do algoritmo proposto e do algoritmo de Chanas e Kolodziejczyk, percebe-se que são parecidas, sendo que a primeira expressa a solução do algoritmo proposto, executado para $H = 100$, e também o fluxo

$G[v_1, v_0] = [18, 20]$, fornecido pelo usuário (neste caso fornecido por Chanas e Kolodziejczyk). Já a segunda figura expressa os resultados do algoritmo de Chanas e Kolodziejczyk, sendo os limitantes do politopo formado pelas capacidades dos cortes-mínimos e o mesmo fluxo $G[v_1, v_0] = [18, 20]$ da Figura 5.5. Verificando o máximo das intersecções entre as soluções dos algoritmos com o fluxo $G[v_1, v_0]$ obteve-se o valor $v = 18,93$, com pertinência 0,467. Em conclusão, percebe-se que os algoritmos possuem as mesmas soluções, porém o algoritmo proposto possui a vantagem de que o usuário não necessita fornecer o fluxo máximo desejável, $G[v_1, v_0]$, como mostram as Tabelas 5.1 e 5.2, o que é muito significativo quando a rede em questão for de grande porte.

Vale ressaltar que o fluxo $G[v_1, v_0]$, contido na Figura 5.5, foi incluído somente para comparar com os resultados do algoritmo de Chanas e Kolodziejczyk, visto que, o mesmo não é necessário.

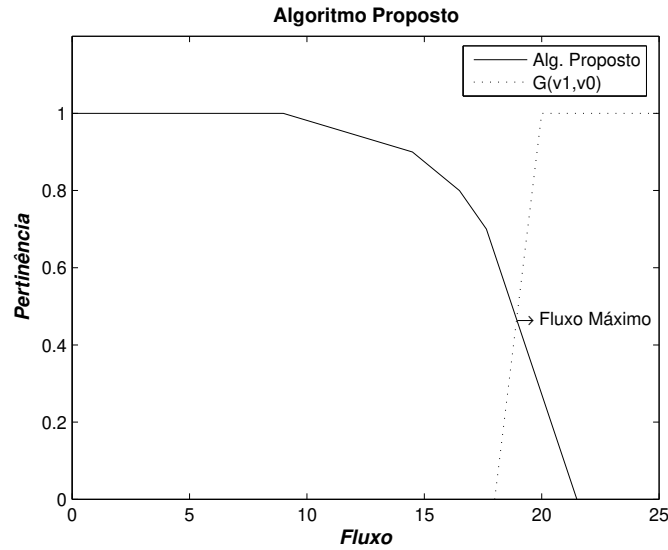


Figura 5.5: Resultados - algoritmo proposto

5.5 Considerações finais

Foi proposto neste capítulo um algoritmo para o problema de fluxo máximo em redes com incertezas nas capacidades (Hernandes, Yamakami, Takahashi e Verdegay 2006). Foi implementada uma heurística, baseada em α -cortes, para tratar das incertezas e utilizou-se a

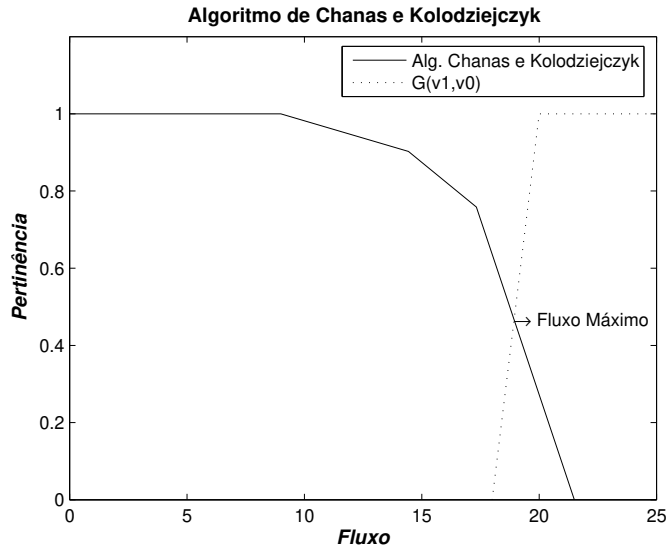


Figura 5.6: Resultados - Chanas e Kolokziejczyk

técnica dos caminhos aumentantes para resolvê-lo.

O algoritmo possui como vantagem a generalidade do conjunto solução, ou seja, o usuário poderá optar por um conjunto solução com vários intervalos, com isso é possível verificar o comportamento do fluxo à medida em que α aumenta. O usuário também poderá optar por solução única. Além desta vantagem, o algoritmo apresenta na solução o comportamento do fluxo ao longo dos arcos da rede, ao contrário dos trabalhos da literatura que somente fornecem o valor do fluxo máximo e os nós que pertencem a cada corte mínimo.

Além disso, nos algoritmos existentes é solicitado que o usuário forneça o valor do fluxo máximo desejável ($G(v_1, v_0)$) e, a partir de então, os algoritmos encontram o grau de satisfação daquele fluxo na rede. Já no algoritmo proposto o usuário não necessita apresentar o fluxo $G(v_1, v_0)$, pois este encontra o fluxo máximo para cada α . Vale ressaltar que quando a rede a ser considerada for de grande porte esta vantagem é bastante considerável, pois fica difícil ao usuário fornecer o fluxo máximo desejável $G(v_1, v_0)$.

Capítulo 6

O Problema de Fluxo de Custo Mínimo com Incertezas

6.1 Introdução

O problema de fluxo de custo mínimo (PFCM) é um outro problema bastante estudado em grafos. Este problema consiste em atender, a um custo mínimo, a demanda em uma rede, dada a oferta de recursos e as restrições de capacidade dos arcos. É um problema utilizado principalmente nas áreas de engenharia e de computação, podendo ser aplicado em problemas tais como: distribuição em redes de TV a cabo; transportes; comunicações; dentre outros (Ahuja et al. 1993, Lawler 1976, Philips e Garcia-Diaz 1981).

Normalmente existem restrições de fluxo associadas a cada arco (máximo e mínimo) e é associado um valor (custo) referente ao trânsito a cada arco, por unidade de fluxo (c_{ij}).

Em muitas aplicações reais, tanto os custos dos arcos quanto as capacidades são parâmetros conhecidos aproximadamente, muitas vezes devido a informações insuficientes. Por exemplo, na avaliação da programação do desenvolvimento de um projeto de pesquisa, o tempo requerido a cada atividade do projeto pode ser um dado incerto. Neste trabalho, é abordado o problema de fluxo de custo mínimo em redes com parâmetros incertos, que será denotado *PFCM fuzzy*, sendo que as incertezas são tratadas utilizando a teoria dos conjuntos *fuzzy* (Zadeh 1965, Zadeh 1968, Zadeh 1978).

Na literatura, são encontrados poucos artigos que abordam o problema de fluxo de custo mínimo *fuzzy*, onde as incertezas apresentadas estão nos custos e/ou nas capacidades. O

principal trabalho é o de Shih e Lee (1999), onde os autores fazem uma adaptação do método Húngaro, utilizando programação possibilística, transformando o problema em um de programação linear *crisp*, sendo que a rede abordada possui custos e capacidades *fuzzy*. Já em outro trabalho da literatura, proposto por Liu e Kao (2004), é utilizado o índice de defuzzificação de Yager (1981), transformando os custos *fuzzy* em *crisp*. Em seguida é aplicado um algoritmo clássico para encontrar a solução do problema. Os trabalhos de Takahashi (2004) figuram entre os mais recentes, onde são propostos três algoritmos para o PFCM *fuzzy*, um considerando apenas o problema com capacidade incerta, outro abordando o custo incerto e o último considerando o custo e a capacidade incertos, sendo este a união dos dois primeiros algoritmos. O algoritmo de Takahashi para o primeiro caso é baseado em α -cortes, resolvendo um problema *crisp* para cada α . Já o segundo resume-se em encontrar soluções factíveis, tendo como idéia trabalhar com as capacidades dos arcos para desviar o fluxo para rotas alternativas, sendo o objetivo enumerar todas as possíveis soluções. O terceiro algoritmo é a união dos dois primeiros, porém, podendo ser aplicado apenas a problemas de pequeno porte devido à estrutura apresentada, visto que é necessário enumerar várias soluções possíveis.

Este trabalho tem como finalidade encontrar soluções factíveis para o PFCM em uma rede com custos e capacidades incertas. São propostos alguns algoritmos para resolver este problema, sendo que o primeiro é uma adaptação do método do Big-M, transformando o problema incerto em um clássico. Os demais algoritmos propostos trabalham com o problema incerto. Para este caso são propostos quatro algoritmos, sendo o primeiro para o PFCM *fuzzy* com um nó origem e um nó destino, o segundo para um nó origem e múltiplos nós destinos, o terceiro para múltiplos nós origens e um nó destino e o quarto para múltiplos nós origens e múltiplos nós destinos.

Este capítulo está organizado da seguinte forma: na Seção 2 são apresentadas as formulações do PFCM clássico e do PFCM *fuzzy*. Os algoritmos propostos estão na Seção 3. Na Seção 4 encontram-se os resultados computacionais. Na Seção 5 são feitos alguns comentários.

6.2 Formulação do problema de fluxo de custo mínimo (PFCM)

Nesta seção são formulados os problemas de fluxo de custo mínimo em grafos com e sem incertezas nos custos e nas capacidades dos arcos.

6.2.1 Formulação do PFCM clássico

Seja $G = (V, A)$ uma rede direcionada com custo c_{ij} associado a cada unidade de fluxo x_{ij} que percorre o arco $(i, j) \in A$. Para cada arco (i, j) tem-se uma capacidade mínima l_{ij} e máxima u_{ij} de fluxo, e para cada nó tem-se um parâmetro b_i , tal que:

- $b_i > 0$, se i for um nó gerador (nó-origem);
- $b_i = 0$, se i for um nó transbordo; e
- $b_i < 0$, se i for um nó consumidor (nó-destino).

O problema de fluxo de custo mínimo clássico é formulado da seguinte forma:

$$\begin{aligned} \min z = & \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s.a :} & \sum_{j=1}^n x_{ij} - \sum_{j=1}^n x_{ji} = b_i \quad i = 1, 2, \dots, n \\ & l_{ij} \leq x_{ij} \leq u_{ij} \quad (i, j) \in A \end{aligned} \tag{6.1}$$

Existem vários algoritmos desenvolvidos para a resolução do PFCM. Dentre os mais conhecidos destacam-se: o método simplex para redes e o algoritmo *Out-of-Kilter* (Bazarrá et al. 1990).

6.2.2 Formulação do PFCM *fuzzy*

Suponha que se tenha um PFCM com custos e capacidades incertos. Este problema é formulado da seguinte forma:

$$\begin{aligned} \min z = & \sum_{i=1}^n \sum_{j=1}^n \tilde{c}_{ij} x_{ij} \\ \text{s.a :} & \sum_{j=1}^n x_{ij} - \sum_{j=1}^n x_{ji} = b_i \quad i = 1, 2, \dots, n \\ & \tilde{l}_{ij} \leq x_{ij} \leq \tilde{u}_{ij} \quad (i, j) \in A \end{aligned} \tag{6.2}$$

onde $\tilde{c}_{ij}$, $\tilde{l}_{ij}$, $\tilde{u}_{ij}$ representam o custo *fuzzy* e os limitantes inferior *fuzzy* e superior *fuzzy* da capacidade, respectivamente.

Sem perda de generalidade os números *fuzzy* adotados são triangulares e o limitante inferior da capacidade é $(0,0,0)$.

6.3 Algoritmos propostos

Nesta seção são apresentados os algoritmos propostos, sendo que estes são abordados utilizando duas técnicas, a primeira é a de transformar o problema incerto em um problema *crisp*. Para isso é apresentado um algoritmo que é uma adaptação do algoritmo do Big-M. Já a segunda técnica não trabalha com índices de defuzzificação, mas sim com o problema incerto.

6.3.1 Adaptação do método do Big-M

Considerando que em alguns problemas as incertezas das capacidades são parecidas, isto é, possuem espalhamentos semelhantes, ou considerando que o usuário deseja obter uma solução rápida e aproximada, foi adaptado o método do Big-M, que transforma o problema com incertezas em um problema clássico.

Os trabalhos encontrados na literatura utilizam índices de defuzzificação ou aplicam α -cortes para transformar o problema em um *crisp* e muitas vezes utilizam técnicas que podem exigir uma implementação bastante elaborada. Considerando este aspecto, é proposto um algoritmo que utiliza o método clássico do Big-M, que é um método simples e bastante utilizado.

No algoritmo adaptado as incertezas nos custos e nas capacidades são tratadas de formas independentes. Para as capacidades são aplicados α -cortes, para assim encontrar soluções com graus de pertinência iguais a α , como proposto por Takahashi (2004). Já os custos são abordados de duas maneiras, sendo que na primeira é aplicado o primeiro índice de defuzzificação de Yager (1981) (centróide), semelhante ao que foi feito no trabalho de Liu e Kao (2004) e, na outra maneira, é utilizado o valor modal. Cabe lembrar que se pode resolver

este problema usando qualquer índice de defuzzificação da literatura, ficando a critério do usuário. A seguir é apresentado o algoritmo adaptado.

Informações sobre o algoritmo:

h : número de partições do intervalo $[0, 1]$;

α_k : valor de pertinência da solução F_k ;

$\tilde{u}_{ij}$: capacidade superior (número trapezoidal *fuzzy*) do arco (i, j) (Seção ??);

$\tilde{c}_{ij}$: custo (número triangular *fuzzy*) do arco (i, j) ; e

$f(\tilde{c}_{ij})$: custo do arco (i, j) defuzzificado, ou seja, após aplicado o índice de defuzzificação.

Algoritmo Proposto

PASSO 1 Defina o número de partições do intervalo h . Para cada $\alpha_k = \frac{k}{h}$, $k = 1, 2, \dots, h$, execute os passos seguintes do algoritmo.

PASSO 2 Aplique um índice de defuzzificação nos custos da rede, obtendo $f(\tilde{c}_{ij})$.

PASSO 3 Para cada α_k resolva o problema associado com $f(\tilde{c}_{ij})$ e $u_{ij} = (\tilde{u}_{ij})_\alpha$ aplicando o método do Big-M e conseqüentemente a Fase II (estes métodos são descritos no Apêndice B desta tese).

PASSO 4 Solução encontrada para cada α_k :

$$F_k \leftarrow F.$$

6.3.2 Algoritmos propostos para problemas com incertezas

Nesta seção são apresentados os algoritmos propostos para resolver o PFCM *fuzzy*. Estes algoritmos, diferentes dos existentes na literatura, não utilizam índices de defuzzificação para transformar o problema com incerteza em um problema clássico. Vale ressaltar que não se tem conhecimento, na literatura, de algoritmos que tratem deste problema sem o transformarem em um problema *crisp*.

Os algoritmos propostos são apresentados a seguir, sendo que estes abordam todas as

possíveis situações, a saber: PFCM com um nó origem e um nó destino (PFCM-UOUD); PFCM com um nó origem e múltiplos nós destinos (PFCM-UOMD); PFCM com múltiplos nós origens e um nó destino (PFCM-MOUD) e PFCM com múltiplos nós origens e múltiplos nós destinos (PFCM-MOMD). Cada um destes algoritmos são descritos a seguir.

6.3.2.1 Algoritmo para o PFCM-UOUD

Este algoritmo utiliza o índice de possibilidade de Dubois e Prade (1983) para ordenar os custos dos caminhos não dominados. A ordenação é feita da seguinte forma: após determinados todos os caminhos não dominados entre os nós origem e destino é verificada, através da medida de possibilidade de Dubois e Prade (detalhes, Capítulo 2, Seção 2.6), qual a possibilidade de cada um destes ter o custo menor do que todos os demais (Exemplo 6.1). Após encontradas as possibilidades dos custos dos caminhos não dominados é verificada a pertinência das capacidades de cada caminho, sendo estas são determinadas da seguinte forma:

$$\mu_{capac_{p_k}} = \min_{l \in P_k} \{\mu_{capac}(l)\} = \text{Poss}\{\text{fluxo passar em } p_k\} \text{ (Exemplo 6.2).}$$

A seguir são apresentados dois exemplos para ilustrar os comentários anteriores.

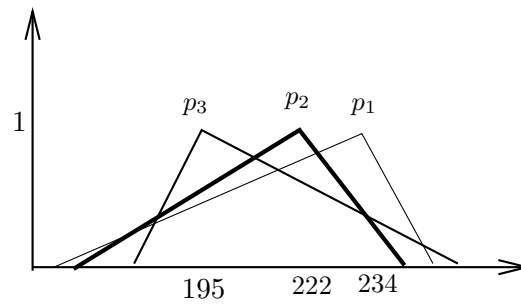
Exemplo 6.1 Ordenação dos caminhos

Dados três caminhos (p_1, p_2, p_3) não dominados com seus respectivos custos na forma (m, α, β) : $(234, 75, 15)$, $(222, 62, 13)$ e $(195, 18, 61)$, ordená-los utilizando a medida de possibilidade de Dubois e Prade (1983).

Solução:

- $\text{Poss}(p_1 < p_2) = 0,8636$, $\text{Poss}(p_1 < p_3) = 0,7132 \rightarrow \text{Poss}\{p_1 \text{ ser mínimo}\} = 0,7132$
- $\text{Poss}(p_2 < p_1) = 1$, $\text{Poss}(p_2 < p_3) = 0,7805 \rightarrow \text{Poss}\{p_2 \text{ ser mínimo}\} = 0,7805$
- $\text{Poss}(p_3 < p_1) = 1$, $\text{Poss}(p_3 < p_2) = 1 \rightarrow \text{Poss}\{p_3 \text{ ser mínimo}\} = 1$

Logo, os caminhos ficam ordenados da seguinte forma: p_3 , p_2 e p_1 (Figura 6.1).

Figura 6.1: Ordenação dos caminhos p_1 , p_2 , p_3 **Exemplo 6.2** Pertinência das capacidades dos caminhos

Dado que um caminho p entre os nós 1 e 6 (Figura 6.2) percorre os arcos (1,2), (2,5) e (5,6), e considerando que são percorridos 8,439 unidades de fluxo nestes arcos, qual é a pertinência da capacidade de p (μ_{capac_p})?

As capacidades dos arcos da Figura 6.2 estão expressas na Tabela 6.1, sendo escritas na forma: (limitante inferior, extremo modal inferior, extremo modal superior, limitante superior).

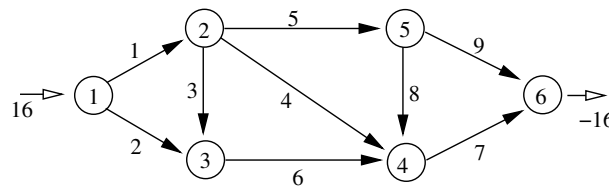


Figura 6.2: Rede exemplo

Arcos	Capacidades
1	(0 0 8 10)
2	(0 0 12 16)
3	(0 0 20 22)
4	(0 0 11 18)
5	(0 0 16 18)
6	(0 0 10 15)
7	(0 0 15 20)
8	(0 0 16 18)
9	(0 0 23 27)

Tabela 6.1: Capacidades dos arcos da Figura 6.2

Solução:

Analisando as capacidades dos arcos do caminho tem-se:

- arco (1,2) possui capacidade (0 0 8 10), calculando a pertinência deste arco tem-se 0,7805 (Figura 6.3);
- arco (2,5) possui capacidade (0 0 16 18), verifica-se que a quantidade de fluxo que percorre o arco é menor que 16 (extremo superior do modal), logo a pertinência deste arco é 1; e
- arco (5,6) idem arco (2,5).

Logo, $\mu_{capac}(p) = \min_{l \in p} \{\mu_{capac}(l)\} = \min\{0.7805; 1; 1\} = 0.7805$.

Conforme ilustra a Figura 6.3, a capacidade do arco (1,2) é um número trapezoidal fuzzy expresso da seguinte forma:

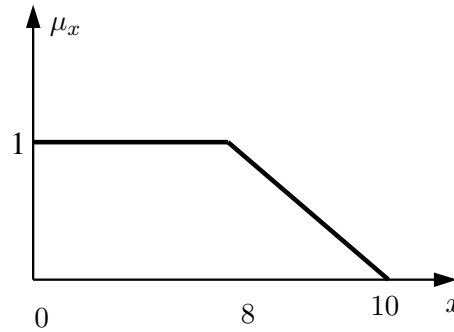


Figura 6.3: Capacidade do arco (1,2)

A seguir é apresentado o algoritmo proposto.

Informações sobre o algoritmo:

Pod : conjunto de todos os caminhos não dominados entre os nós origem e destino;

p_k : k -ésimo caminho não dominado entre os nós origem e destino;

α : pertinência mínima exigida para o fluxo percorrer na rede; e

μ_{p_k} : possibilidade do caminho p_k ser menor do que todos os outros.

Algoritmo proposto para uma origem e um destino

PASSO 1 Encontrar o conjunto de caminhos não-dominados (Pod) com possibilidade de serem mínimos ($\geq \alpha$).

PASSO 2 Ordenar os caminhos $p_k \in Pod$ da seguinte forma:

$$\mu_{p_k} = Poss\{p_k \text{ ser o melhor caminho}\} =$$

$$= \min_{p_k} \{\mu_{custo_{p_k}}, \mu_{capac_{p_k}}\}$$

tal que:

- $\mu_{custo_{p_k}} = Poss\{p_k \text{ ser mínimo}\}$; e
- $\mu_{capac_{p_k}} = \min_{l \in P_k} \{\mu_{capac}(l)\} = Poss\{\text{fluxo passar em } p_k\}$.

PASSO 3 Enviar fluxo para o melhor caminho (em termos de μ_{p_k}) até que:

$$\mu_{p_k \in Pod} = \min\{\mu_{custo_{p_k}}, \mu_{capac_{p_k}}\} = \mu_{p_{k+1} \in Pod} \text{ (próximo caminho)}$$

Caso $\mu_{p_k} = \mu_{p_{k+1}}$, preencher este caminho até o caminho que tenha pertinência diferente.

PASSO 4 Atualizar μ_{p_k} para todo $p_k \in Pod$ e reordená-los baseado em:

$$\mu_{p_k} = \min_{p_k} \{\mu_{custo_{p_k}}, \mu_{capac_{p_k}}\}$$

PASSO 5 Critério de parada.

- se existir fluxo a transitar $\Rightarrow$ Passo 3;
- senão $\Rightarrow$ FIM.

Na seção dos resultados computacionais é feito um exemplo bastante detalhado deste algoritmo.

6.3.2.2 Algoritmo para o PFCM-UOMD ou PFCM-MOUD

Este algoritmo é aplicado ao problema do PFCM *fuzzy* quando a rede a ser considerada possui múltiplas origens e um destino ou uma origem e múltiplos destinos. Quando comparado ao algoritmo anterior (Seção 6.3.2.1) verifica-se que a diferença está no Passo 1, visto que neste

algoritmo é necessário encontrar o conjunto de todos os caminhos não dominados para cada par nó origem-destino. Os demais passos são semelhantes aos da seção anterior.

Algoritmo proposto para várias origens e um destino ou vice-versa

PASSO 1 Encontrar o conjunto de todos os caminhos não-dominados $\{Pod_1, Pod_2, Pod_3, \dots, Pod_n\}$ com possibilidade de serem mínimos ($\geq \alpha$).

PASSO 2 Demais passos: semelhantes aos passos do algoritmo da Seção 6.3.2.1.

6.3.2.3 Algoritmo para o PFCM-MOMD

Este algoritmo é aplicado ao problema do PFCM *fuzzy* em redes com múltiplos nós origens e múltiplos nós destinos. Pode-se verificar que este é um problema difícil, sendo comprovado pela inexistência de trabalhos na literatura.

A principal diferença deste algoritmo com os propostos anteriormente está no Passo 3.

Algoritmo proposto para múltiplas origens e múltiplos destinos

PASSO 1 Encontrar os conjuntos de todos os caminhos não-dominados para cada par de nós origem-destino $\{Pod_1, Pod_2, Pod_3, \dots, Pod_n\}$ com possibilidade de serem mínimos ($\geq \alpha$).

PASSO 2 Ordenar todos os caminhos $p_k \in \{Pod_1, Pod_2, Pod_3, \dots, Pod_n\}$:

$$\mu_{p_k} = Poss\{p_k \text{ ser o melhor caminho}\} =$$

$$= \min_{p_k} \{\mu_{custo_{p_k}}, \mu_{capac_{p_k}}\}$$

tal que:

- $\mu_{custo_{p_k}} = Poss\{p_k \text{ ser mínimo}\}$; e
- $\mu_{capac_{p_k}} = \min_{l \in P_k} \{\mu_{capac}(l)\} = Poss\{\text{fluxo passar em } p_k\}$.

PASSO 3 Para o nó destino i_1 do primeiro caminho ordenado, executar os seguintes passos:

- Enviar fluxo para o melhor caminho (em termos de μ_{p_k}) até que:

$$\mu_{p_k \in Pod} = \min\{\mu_{custo_{p_k}}, \mu_{capac_{p_k}}\} = \mu_{p_{k+1}} \in \{Pod_1, Pod_2, Pod_3, \dots, Pod_n\} \text{ (próximo caminho)}$$

Caso $\mu_{p_k} = \mu_{p_{k+1}}$, preencher este caminho até o caminho que tenha pertinência diferente.

- Atualizar μ_{p_k} para todo $p_k \in \{Pod_1, Pod_2, Pod_3, \dots, Pod_n\}$ e reordená-los baseado em:

$$\mu_{p_k} = \min_{p_k} \{\mu_{custo_{p_k}}, \mu_{capac_{p_k}}\}$$

- Se o fluxo necessário ao nó-destino i_1 já foi satisfeito, eliminar todos os caminhos não dominados que tenham como nó destino i_1 e voltar ao Passo 2 do algoritmo. Caso contrário, retorne à primeira etapa deste passo.

PASSO 4 Critério de parada.

- se existir fluxo a transitar $\Rightarrow$ Passo 3;
- senão $\Rightarrow$ FIM.

6.4 Resultados computacionais

Os algoritmos propostos foram implementados em Matlab 7.0 e executados em algumas redes.

Ressalta-se que os valores das capacidades dos arcos estão expressos na forma $(m_1 - \delta, m_1, m_2, m_2 + \beta)$, sendo que $m_1 - \delta = m_1 = 0$. Já os custos estão denotados na forma padrão adotada, ou seja, (m, α, β) .

Os resultados são discriminados em duas seções, na primeira os obtidos utilizando a adaptação do Big-M e na segunda, os resultados dos demais algoritmos propostos.

6.4.1 Resultados do algoritmo adaptado do Big-M

Este algoritmo foi aplicado na rede da Figura 6.4, tendo os nós 1 e 2 como origens e os nós 4 e 6 como destinos.

Os custos e as capacidades desta estão representados na Tabela 6.2.

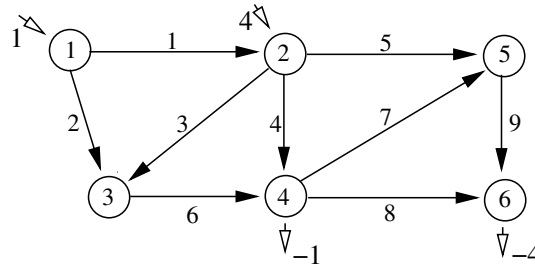


Figura 6.4: Rede PFCM-Clássico

Arcos	Custos	Capacidades
1	(2 1 1)	(0 0 3 4)
2	(5 1 1)	(0 0 1,5 2,5)
3	(3 3 2)	(0 0 2 3)
4	(6 4 2)	(0 0 2 4)
5	(5 1 1)	(0 0 7 8)
6	(6 1 1)	(0 0 3 4)
7	(2 1 1)	(0 0 4 6)
8	(5 4 2)	(0 0 4 5)
9	(3 1 1)	(0 0 2 3)

Tabela 6.2: Custos e capacidades da Figura 6.4

O intervalo $[0,1]$ foi dividido em cinco subintervalos, sendo os valores de α expressos na Tabela 6.3.

Arcos/capac.	$\alpha = 0$	$\alpha = 0.2$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$	$\alpha = 1$
1	4,0	3,8	3,6	3,4	3,2	3,0
2	2,5	2,3	2,1	1,9	1,7	1,5
3	3,0	2,8	2,6	2,4	2,2	2,0
4	4,0	3,6	3,2	2,8	2,4	2,0
5	8,0	7,8	7,6	7,4	7,2	7,0
6	4,0	3,8	3,6	3,4	3,2	3,0
7	6,0	5,6	5,2	4,8	4,4	4,0
8	5,0	4,8	4,6	4,4	4,2	4,0
9	3,0	2,8	2,6	2,4	2,2	2,0

Tabela 6.3: Valores dos α_k

As Tabelas 6.4 e 6.5 representam, respectivamente, os custos defuzzificados e os resultados da rede para cada α e cada custo.

Arcos	Custos (Yager)	Custos (Valor modal)
1	2	2
2	5	5
3	2,6667	3
4	5,3333	6
5	5	5
6	6	6
7	2	2
8	4,3333	5
9	3	3

Tabela 6.4: Valores dos custos defuzzificados

Arcos/ Índice	Yager $\alpha = 0$	Modal $\alpha = 0$	Yager $\alpha = 0,2$	Modal $\alpha = 0,2$	Yager $\alpha = 0,4$	Modal $\alpha = 0,4$	Yager $\alpha = 0,6$	Modal $\alpha = 0,6$	Yager $\alpha = 0,8$	Modal $\alpha = 0,8$	Yager $\alpha = 1$	Modal $\alpha = 1$
1	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	0,6	1,0	0,0
2	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,4	0,0	1,0
3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,4	0,0	1,0	0,0
4	2,0	2,0	2,2	2,2	2,4	2,4	2,6	2,6	2,4	2,4	2,0	2,0
5	3,0	3,0	2,8	2,8	2,6	2,6	2,4	2,4	2,2	2,2	2,0	2,0
6	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,4	0,4	1,0	1,0
7	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
8	1,0	1,0	1,2	1,2	1,4	1,4	1,6	1,6	1,8	1,8	2,0	2,0
9	3,0	3,0	2,8	2,8	2,6	2,6	2,4	2,4	2,2	2,2	2,0	2,0
Custo	41	43	41,33	43,6	41,66	44,2	42	44,8	43,66	46,6	46	49

Tabela 6.5: Fluxo final

Analisando os resultados percebe-se que os mesmos são parecidos, sendo diferenciados apenas para $\alpha = 0.8$ e $\alpha = 1.0$. Mas, através dos custos conclui-se que, para este exemplo, o índice de defuzzificação de Yager é melhor do que o modal.

6.4.2 Resultados do PFCM-UOD

O algoritmo proposto foi executado na rede da Figura 6.5, tendo como nó origem o nó 1 e destino o nó 6.

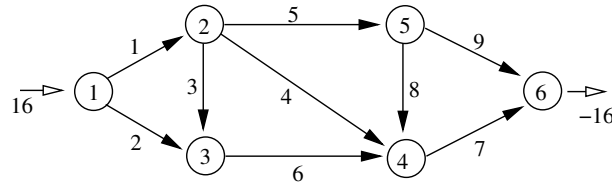


Figura 6.5: Rede PFCM-uma entrada e uma saída

Arcos	Custos	Capacidades
1	(45 12 5)	(0 0 8 10)
2	(57 15 4)	(0 0 12 16)
3	(52 2 9)	(0 0 20 22)
4	(79 28 6)	(0 0 11 18)
5	(58 2 14)	(0 0 16 18)
6	(55 12 5)	(0 0 10 15)
7	(110 35 4)	(0 0 15 20)
8	(40 8 6)	(0 0 16 18)
9	(92 4 42)	(0 0 23 27)

Tabela 6.6: Custos e capacidades da Figura 6.5

Aplicando o algoritmo proposto (Algoritmo 6.3.2.1) passo a passo tem-se:

1ª iteração:

PASSO 1 Seja $\alpha = 0.7$. Encontrar os caminhos não-dominados entre os nós 1 e 6 com possibilidade superior a 0.7.

Caminhos encontrados (Pod):

Caminho 1 (p_1): $1 \rightarrow 2 \rightarrow 4 \rightarrow 6$ com custo (234, 75, 15).

Caminho 2 (p_2): $1 \rightarrow 3 \rightarrow 4 \rightarrow 6$ com custo (222, 62, 13).

Caminho 3 (p_3): $1 \rightarrow 2 \rightarrow 5 \rightarrow 6$ com custo (195, 18, 61).

PASSO 2 Ordenar os caminhos: $p_k \in Pod$

$$\mu_{p_k} = Poss\{p_k \text{ ser o melhor caminho}\} =$$

$$= \min_{p_k} \{\mu_{custo_{p_k}}, \mu_{capac_{p_k}}\}$$

tal que:

- $\mu_{custo_{p_k}} = Poss\{p_k \text{ ser mínimo}\}$; e
- $\mu_{capac_{p_k}} = \min_{l \in P_k} \{\mu_{capac}(l)\} = Poss\{\text{fluxo passar em } p_k\}$.

$$- Poss(p_1 < p_2) = 0,8636, Poss(p_1 < p_3) = 0,7132$$

$$- Poss(p_2 < p_1) = 1, Poss(p_2 < p_3) = 0,7805$$

$$- Poss(p_3 < p_1) = 1, Poss(p_3 < p_2) = 1$$

- Ordenação: p_3, p_2 e p_1

$$\mu_{p_1} = \min\{0,7132; 1\} = 0,7132$$

$$\mu_{p_2} = \min\{0,7805; 1\} = 0,7805$$

$$\mu_{p_3} = \min\{1; 1\} = 1$$

PASSO 3 Enviar fluxo para o 1º caminho até alcançar a pertinência do caminho seguinte:

$$\mu_{p_3 \in Pod} = \min\{\mu_{custo_{p_3}}, \mu_{capac_{p_3}}\} = \mu_{p_2 \in Pod} \text{ (próximo caminho)}$$

Relaxar as capacidades dos arcos do caminho p_3 até a pertinência de 0,7805 (ordenado segundo melhor caminho).

Fluxo ao longo dos arcos é: (8,439 0 0 0 8,439 0 0 0 8,439)

PASSO 4 Atualizar μ_{p_k} para todo $p_k \in Pod$ baseado em

$$\min_{p_k} \{\mu_{custo_{p_k}}, \mu_{capac_{p_k}}\}$$

- Para o caminho p_1 :

$$\mu_{p_1} = \min_{p_1} \{0, 7132; 0, 7805\} = 0, 7132$$

tal que:

$$- \mu_{capac}(p_1) = \min_{l \in P_1} \{\mu_{capac}(l)\} = \min\{0, 7805; 1; 1\} = 0, 7805$$

$$- \mu_{custo}(p_1) = 0, 7132 \text{ (já foi calculado anteriormente, não muda)}$$

- Para o caminho p_2 :

$$\mu_{p_2} = \min_{p_2} \{1; 0, 7805\} = 0, 7805$$

tal que:

$$- \mu_{capac}(p_2) = \min_{l \in P_2} \{\mu_{capac}(l)\} = \min\{1; 1; 1\} = 1$$

$$- \mu_{custo}(p_2) = 0, 7805 \text{ (já foi calculado anteriormente, não muda)}$$

- Para o caminho p_3 :

$$\mu_{p_3} = \min_{p_3} \{0, 7805; 1\} = 0, 7805$$

tal que:

$$- \mu_{capac}(p_3) = \min_{l \in P_3} \{\mu_{capac}(l)\} = \min\{0, 7805; 1; 1\} = 0, 7805$$

$$- \mu_{custo}(p_3) = 1 \text{ (já foi calculado anteriormente, não muda)}$$

- Ordenação: p_3 , p_2 e p_1 (quando houver empate entre dois caminhos, será critério de desempate aquele que possuir o menor custo)

PASSO 5 Ainda há fluxo a transitar.

2ª iteração:

PASSO 3 Preencher o 1º caminho até alcançar a pertinência do seguinte:

(Como p_3 (1º caminho) possui a mesma pertinência de p_2 (2º caminho), então terá

que enviar fluxo em p_3 até o caminho p_1 (3º caminho)):

$$\mu_{p_3 \in Pod} = \min\{\mu_{custo_{p_3}}, \mu_{capac_{p_3}}\} = \mu_{p_1 \in Pod} \text{ (próximo caminho)}$$

Relaxar as capacidades dos arcos do caminho p_3 até a pertinência de 0,7132

$$\text{Fluxo} = (8,5735 \ 0 \ 0 \ 0 \ 8,5735 \ 0 \ 0 \ 0 \ 8,5735)$$

PASSO 4 Atualizar μ_{p_k} para todo $p_k \in Pod$ baseado em

$$\min_{p_k} \{\mu_{custo_{p_k}}, \mu_{capac_{p_k}}\}$$

- Para o caminho p_1 :

$$\mu_{p_1} = \min_{p_1} \{0, 7132; 0.7132\} = 0, 7132$$

tal que:

$$- \mu_{capac}(p_1) = \min_{l \in P_1} \{\mu_{capac}(l)\} = \min\{0, 7132; 1; 1\} = 0, 7132$$

$$- \mu_{custo}(p_1) = 0, 7132 \text{ (já foi calculado anteriormente, não muda)}$$

- Para o caminho p_2 :

$$\mu_{p_2} = \min_{p_2} \{1; 0, 7805\} = 0, 7805$$

tal que:

$$- \mu_{capac}(p_2) = \min_{l \in P_2} \{\mu_{capac}(l)\} = \min\{1; 1; 1\} = 1$$

$$- \mu_{custo}(p_2) = 0, 7805 \text{ (já foi calculado anteriormente, não muda)}$$

- Para o caminho p_3 :

$$\mu_{p_3} = \min_{p_3} \{0, 7132; 1\} = 0, 7132$$

tal que:

$$- \mu_{capac}(p_3) = \min_{l \in P_3} \{\mu_{capac}(l)\} = \min\{0, 7132; 1; 1\} = 0, 7132$$

$$- \mu_{custo}(p_3) = 1 \text{ (já foi calculado anteriormente, não muda)}$$

- Ordenação: p_2, p_3 e p_1

PASSO 5 Ainda há fluxo a transitar.

3ª iteração:

PASSO 3 Enviar fluxo para o 1º caminho até alcançar a pertinência do seguinte:

$$\mu_{p_2 \in Pod} = \min\{\mu_{custo_{p_2}}, \mu_{capac_{p_2}}\} = \mu_{p_3 \in Pod} \text{ (próximo caminho)}$$

Relaxar as capacidades dos arcos do caminho p_3 até a pertinência de 0,7132

$$\text{Fluxo} = (8,5735 \ 7,4265 \ 0 \ 0 \ 8,5735 \ 7,4265 \ 7,4265 \ 0 \ 8,5735)$$

PASSO 4 Similar ao da 2ª iteração, que não será calculado porque o programa será finalizado.

PASSO 5 FIM

Portanto, a **solução final** é:

Arco	1	2	3	4	5	6	7	8	9
Fluxo final	8,5735	7,4265	0	0	8,5735	7,4265	7,4265	0	8,5735

com pertinência 0,7132 e custo final: (332,05 61,48 61,95).

O custo final é calculado através da multiplicação dos custos dos arcos com a quantidade de fluxo que percorre em cada arco.

6.4.3 Resultados do PFCM-UOMD

O algoritmo proposto foi executado na rede da Figura 6.6, tendo como nó origem o nó 1 e destinos os nós 4, 5 e 6. Na Tabela 6.7 estão os dados da rede.

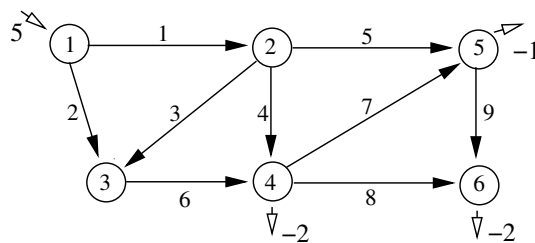


Figura 6.6: Rede PFCM-uma entrada e múltiplas saídas

Considerando $\alpha = 0.7$ a solução final é:

Arco	1	2	3	4	5	6	7	8	9
Fluxo final	3,2727	1,7273	0	2,0227	1,25	1,7273	0	1,75	0,25

com pertinência 0,7273 e custo final: (48,25 26,7727 13,7)

Arcos	Custos	Capacidades
1	(2 1 1)	(0 0 3 4)
2	(3 2 1)	(0 0 1,5 2,5)
3	(3 3 2)	(0 0 2 3)
4	(6 4 2)	(0 0 2 4)
5	(5 1 1)	(0 0 7 8)
6	(5 2 1)	(0 0 3 4)
7	(2 1 1)	(0 0 4 6)
8	(5 4 2)	(0 0 4 5)
9	(3 1 1)	(0 0 2 3)

Tabela 6.7: Custos e capacidades da Figura 6.6

6.4.4 Resultados do PFCM-MOUD

O algoritmo proposto foi executado na rede da Figura 6.7, tendo como nós origens 1, 3 e 7 e destino o nó 6.

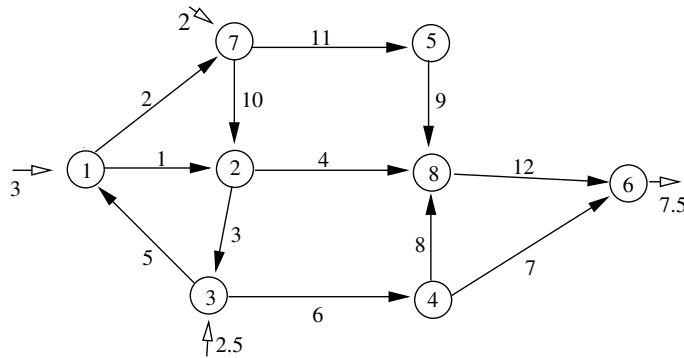


Figura 6.7: Rede PFCM-múltiplas entradas e uma saída

Arcos	Custos	Capacidades
1	(6 2 1)	(0 0 2,5 3,5)
2	(5 1 2)	(0 0 2,5 3,5)
3	(3 1 1)	(0 0 2 3)
4	(8 2 1)	(0 0 2 3)
5	(2,5 0,5 0,5)	(0 0 1 2)
6	(4,5 1,5 1,5)	(0 0 3 4)
7	(4 1 1)	(0 0 2,5 4)
8	(1,5 0,5 1,5)	(0 0 2 3)
9	(5 2 1)	(0 0 2,5 4)
10	(4,5 2 2)	(0 0 4 5)
11	(3 1 1)	(0 0 3 5)
12	(2 1 1)	(0 0 5 6)

Tabela 6.8: Custos e capacidades da Figura 6.7

Considerando $\alpha = 0.7$ a solução final é:

Arco	1	2	3	4	5	6	7	8	9	10	11	12
Fluxo final	2,1	0,9	0	2,1	0	2,5	2,5	0	2,9	0	2,9	5

com pertinência 0,7333 e custo final: (88,35 29,25 23,05)

6.4.5 Resultados do PFCM-MOMD

O algoritmo proposto foi executado na rede da Figura 6.8, tendo como nós origens os nós 1 e 2 e destinos os nós 4, 5 e 6.

Para ilustração, o α exigido inicialmente foi 0.05, ou seja, uma pertinência relativamente baixa.

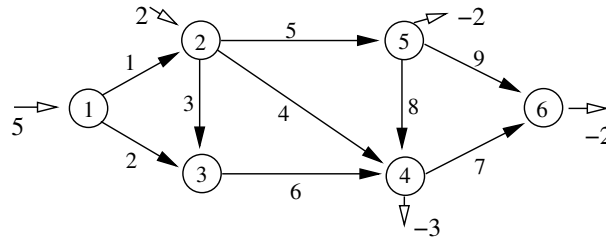


Figura 6.8: Rede PFCM-múltiplas entradas e múltiplas saídas

Ressalta-se que os custos e as capacidades são os mesmos da Tabela 6.7.

Os resultados obtidos com a execução deste algoritmo são:

Arco	1	2	3	4	5	6	7	8	9
Fluxo final	3,7778	1,2222	0	1,7778	4	1,2222	0	0	2

com pertinência 0,1111 e custo final: (54 19,7778 15,7778)

Vale ressaltar que este algoritmo necessita ser testado para redes de grande porte.

6.5 Considerações finais

Problemas de redes têm sido extensamente estudados e aplicados nas soluções de problemas reais. Este trabalho apresenta alguns algoritmos aplicados a problemas de fluxo de custo mínimo com relaxamento nas exigências de valores precisos para os parâmetros relacionados aos arcos do grafo, ou seja, com incertezas nos parâmetros. Com exceção dos algoritmos propostos por Shih e Lee (1999) e Takahashi (2004), os demais da literatura apresentam incertezas em somente um dos parâmetros. Ao contrário, os algoritmos aqui propostos, além de abordar as incertezas em dois parâmetros (custos e capacidades), possuem a vantagem de serem aplicáveis a problemas de grande porte. Por outro lado, com exceção do primeiro algoritmo, os demais trabalham com o problema na forma *fuzzy*.

Vale ressaltar que, embora os algoritmos propostos sejam baseados no PFCM, suas aplicações não estão restritas apenas a problemas desta categoria, podendo, por exemplo, serem aplicados a problemas de caminho mínimo com restrições de custo e tempos incertos.

Capítulo 7

Coloração em Grafos *Fuzzy*

7.1 Introdução

Coloração em grafos é um problema de otimização combinatória bastante estudado, visto que muitos problemas práticos, tais como: gerência, alocação de recursos e atribuições de frequências, podem ser modelados como deste tipo. A definição clássica deste problema consiste em colorir todos os vértices (nós) de um grafo não orientado utilizando o menor número possível de cores, exigindo que os extremos de cada aresta possuam cores distintas.

Em algumas circunstâncias, entretanto, este esquema de coloração parece ser muito restritivo, visto que a única restrição que o problema pode ter é que nós adjacentes não possuam a mesma cor. Com base nisto, Muñoz et al (Muñoz, Ortuño, Ramírez e Yáñez 2003, Muñoz, Ortuño, Ramírez e Yáñez 2005) e Ramírez (2001) propõem, baseado na relaxação do conceito de incompatibilidade, uma extensão deste problema, surgindo então a *coloração em grafos fuzzy*. Tal extensão se justifica, pois em muitas situações reais a incompatibilidade não é *crisp*, visto que dois nós podem ser mais ou menos incompatíveis, ou a incompatibilidade pode apresentar diferentes graus.

Este capítulo tem como objetivo apresentar as extensões do problema de coloração *fuzzy* propostos na literatura, bem como apresentar uma nova extensão, que será chamada *coloração (d, f, g)-estendida* (ou *(d, f, g)-coloração*) e um algoritmo para resolvê-la.

Os trabalhos de Muñoz et al (Muñoz et al. 2003, Muñoz et al. 2005) e Ramírez (2001) são base para o trabalho proposto neste capítulo, visto que são os únicos existentes na literatura

que tratam deste tema e que abordam o problema da (d, f) -coloração (ou *coloração (d, f) -estendida*).

Este capítulo está organizado da seguinte forma: na Seção 2 são apresentados alguns conceitos preliminares sobre o problema de coloração de grafos clássica. Na Seção 3 é abordado o problema de coloração em grafos *fuzzy*. Na Seção 4 está o problema da coloração (d, f) -estendida, que é a extensão da coloração *fuzzy*, proposta por Muñoz et al. (2005). A coloração (d, f, g) -estendida, proposta neste trabalho, é apresentada na Seção 5. O algoritmo proposto para a extensão (d, f, g) está na Seção 6. Alguns resultados computacionais são apresentados na Seção 7. Na Seção 8 encontram-se as considerações finais.

7.2 Conceitos preliminares

Nesta seção são introduzidos alguns conceitos sobre coloração de grafos clássica, ou seja, coloração em grafos com estrutura bem definida.

Formalmente, o problema de coloração é definido da seguinte forma:

Definição 29 *Dado um grafo $G = (V, A)$, uma função de coloração é uma aplicação*

$$C : V \rightarrow \mathbb{N}$$

sendo $C(i)$ a cor atribuída ao nó $i \in V$, de tal forma que dois nós adjacentes não tenham a mesma cor, isto é, $C(i) \neq C(j)$ se $(i, j) \in A$. Neste caso os nós i e j são ditos incompatíveis e o grafo G é denotado grafo de incompatibilidades.

Definição 30 *Uma k -coloração, C^k , é uma função coloração com não mais do que k diferentes cores:*

$$C^k : V \rightarrow \{1, 2, \dots, k\}$$

Definição 31 *Um grafo é k -colorido se admite uma k -coloração. O valor mínimo de k tal que G é k -colorido é denominado número cromático de G , denotado por $X(G)$.*

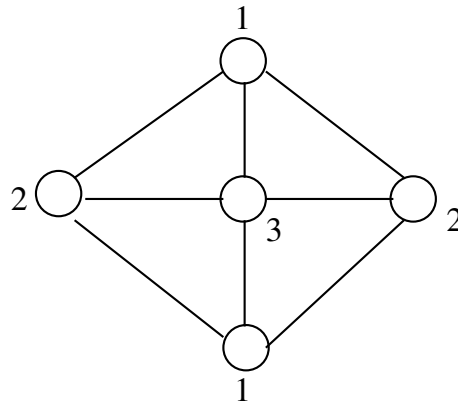


Figura 7.1: Exemplo de uma 3-coloração

A Figura 7.1 exemplifica uma 3-coloração do grafo G .

Logo, o problema de coloração de grafos consiste em determinar o número cromático de um grafo G e uma função de coloração associada. Portanto, o objetivo deste problema é minimizar o número de cores atribuídas aos nós da rede, onde cada nó pode representar certos itens, sendo que nós adjacentes devem ter cores diferentes.

7.3 O problema da coloração *fuzzy*

A fim de introduzir os conceitos da função de coloração de um grafo *fuzzy* e de seu respectivo número cromático, inicialmente é apresentada uma aplicação deste problema.

7.3.1 O problema dos semáforos

O problema dos semáforos consiste em controlar, em um cruzamento, um sistema de luzes de tráfego (conjunto de semáforos) de tal maneira que um determinado nível de segurança seja alcançado. Uma das formas de se resolver este problema é através da coloração de grafos.

Exemplo 7.1 Cruzamento entre duas ruas.

Este exemplo tem como finalidade controlar o tráfego de veículos neste cruzamento, a fim de evitar colisões.

Utilizando as definições de incompatibilidade, comentadas anteriormente, determinadas pistas são compatíveis umas com as outras, tais como AD e CB (Figura 7.2), pois o tráfego

entre estas duas direções pode ocorrer simultaneamente, enquanto que outras são incompatíveis, como AB e CD, sendo que para estas direções o tráfego não poderá ser liberado simultaneamente.

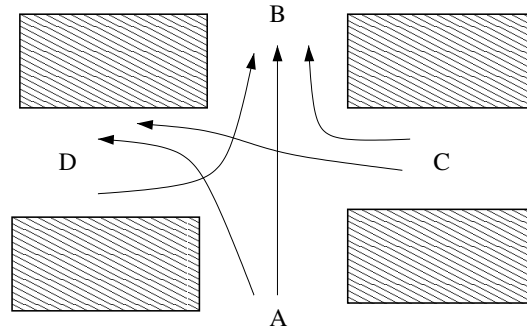


Figura 7.2: Exemplo de um cruzamento entre ruas

Analisando a Figura 7.2 verifica-se que este problema pode ser modelado por meio de um grafo de incompatibilidade, $G = (V, A)$, cujos nós são as ruas (pistas) e um par de pistas define um arco se as mesmas forem incompatíveis, isto é, se estas causarem colisões. Neste caso, tem-se o seguinte grafo (Figura 7.3):

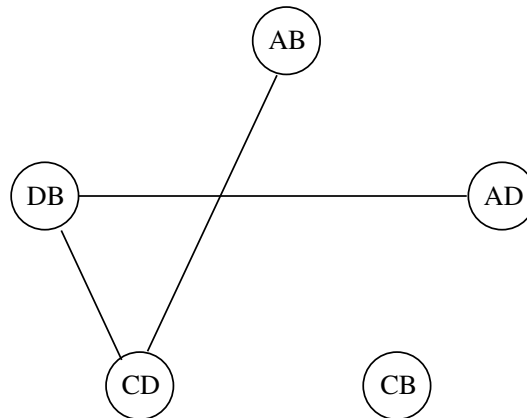


Figura 7.3: Representação da Figura 7.2 em forma de grafo

tal que:

$$V = \{AB, AD, CB, CD, DB\};$$

$$A = \{\{AB, CD\}; \{AD, DB\}; \{CD, DB\}\}.$$

Uma k -coloração, C_k , do grafo G identifica o controle dos semáforos no cruzamento. O ciclo inteiro do sistema de semáforos é dividido em k períodos de tempo.

Resolvendo este problema verifica-se que o número cromático de G é 2 e é uma 2-coloração, logo são necessárias apenas duas fases nos semáforos deste cruzamento, sendo o resultado:

$$C(AB) = 1; C(AD) = 2; C(CB) = 1; C(CD) = 2; C(DB) = 1.$$

Obviamente, a política do controle dos semáforos depende da incompatibilidade das pistas, pois em um horário de muito tráfego não seria conveniente que o trânsito, por exemplo, dos sentidos de direção AD e CD, fossem liberados simultaneamente. Já num horário de pouco movimento isto poderia ocorrer. Logo, o conceito de incompatibilidade poderia ser *fuzzy* e graduado. Esta graduação, que não necessita ser numérica, é associada ao nível desejado de segurança para o tráfego naquele cruzamento, sendo que o nível máximo é alcançado quando todas as direções são consideradas incompatíveis e o grafo é completo; neste caso, o número cromático é o número de direções e a política de segurança dos semáforos é controlar que somente seja permitido um movimento por ciclo. Por outro lado, o nível mínimo de segurança é que todas as direções sejam compatíveis; neste caso, o número cromático é 1 e todos os movimentos são permitidos ao mesmo instante.

No exemplo 7.1 as pistas CD e DB são mais incompatíveis do que as pistas AB e DB. Com isso, pode-se definir o seguinte conjunto de incompatibilidades: seja $I = \{n, b, m, a, t\}$, tal que n, b, m, a e t denotam os graus de incompatibilidade *nulo, baixo, médio, alto* e *total*, respectivamente.

Portanto, este problema pode ser modelado por meio de um grafo *fuzzy* $\tilde{G} = (V, \mu)$, sendo V o conjunto de vértices e μ uma matriz de adjacência que representa as arestas existentes e seus respectivos graus de incompatibilidade.

Por exemplo, seja $V = \{AB, AD, CB, CD, DB\}$; e

$$\mu = \begin{pmatrix} - & n & b & a & b \\ n & - & n & m & a \\ b & n & - & n & n \\ a & m & n & - & a \\ b & a & n & a & - \end{pmatrix}.$$

Este grafo *fuzzy* está descrito na Figura 7.4.

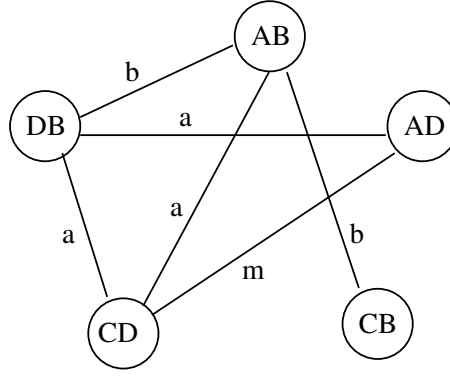


Figura 7.4: Representação do grafo de incompatibilidade do Exemplo 7.1

Um caminho para considerar estas incompatibilidades *fuzzy* é a partir da definição de uma função de coloração para grafos *fuzzy*, conceito este que é introduzido a seguir.

7.3.2 A função coloração de um grafo *fuzzy*

Dado um grafo *fuzzy* $\tilde{G} = (V, \mu)$, uma aproximação natural para obter algum conhecimento sobre isto é analisar uma sequência de α -cortes. Um conjunto *fuzzy* A é definido em X , caracterizado pela família de seus α -cortes

$$A_\alpha = \{x \in X | \mu_A(x) \geq \alpha\}, \alpha \in I$$

Esta família de conjuntos é monótona, isto é, verifica-se que se $\alpha \leq \beta$ então $A_\alpha \supseteq A_\beta$ $\forall \alpha, \beta \in I$.

Por outro lado, (veja Kerre (1993)) dada uma família finita monótona $\{A_{\alpha^p} / p \in \{1, \dots, m\}\}$, um conjunto *fuzzy* A pode ser definido pela função de pertinência $\mu_A(x) = \sup\{\alpha^p / x \in A_{\alpha^p}\}$ $\forall x \in X$.

Seja $G_\alpha = \{(V, A_\alpha) / \alpha \in I\}$ a família do conjunto de α -cortes de $\tilde{G}$, onde o α -corte de um grafo *fuzzy* é um grafo *crisp* $G_\alpha = (V, A_\alpha)$ com $A_\alpha = \{(i, j) | i, j \in V, \mu_{ij} \geq \alpha\}$.

Portanto, uma k -coloração (*crisp*), C_α^k , pode ser definida sobre G_α . A função k -coloração de $\tilde{G}$ é definida através dos α -cortes.

Para cada $\alpha \in I$, dado que X_α denota o número cromático de G_α , o número cromático de $\tilde{G}$ é definido através da família monótona dos conjuntos dos α -cortes.

Definição 32 Dado um grafo *fuzzy* $\tilde{G} = (V, \mu)$, seu número cromático é o número *fuzzy*

$$X(\tilde{G}) = \{(x, v(x)) / x \in X\}$$

tal que: $X = \{1, \dots, |V|\}$, $v(x) = \sup\{\alpha \in I \mid x \in A_\alpha\}$, $\forall x \in X$ e $A_\alpha = \{1, \dots, X_\alpha\} \forall \alpha \in I$.

O número cromático de um grafo *fuzzy* é um número *fuzzy* normalizado cujo valor modal é associado ao conjunto de arestas do grafo. Seu significado depende do sentido do índice α , sendo que ele pode ser interpretado da seguinte maneira: para valores pequenos de α há muitas ligações incompatíveis entre os nós e, conseqüentemente, mais cores são necessárias para considerar estas incompatibilidades; por outro lado, para α com valores maiores há poucas ligações incompatíveis entre os nós e com isso menos cores são necessárias. O número cromático resume todas estas informações a fim de controlar o problema *fuzzy*.

Logo, o *problema de coloração fuzzy* consiste em determinar o número cromático de um grafo *fuzzy* e uma função de coloração associada.

No Exemplo 7.1, cinco grafos *crisp* $G_\alpha = (V, A_\alpha)$ são obtidos considerando os valores α , para cada $\alpha \in I$. A tabela 7.1 contém o conjunto aresta A_α , o número cromático X_α e uma X_α -coloração, $C_\alpha^{X_\alpha}$.

α	A_α	X_α	$C_\alpha^{X_\alpha}(AB)$	$C_\alpha^{X_\alpha}(AD)$	$C_\alpha^{X_\alpha}(CB)$	$C_\alpha^{X_\alpha}(CD)$	$C_\alpha^{X_\alpha}(DB)$
n	$\{AB, AD\}; \{AB, CB\}; \{AB, CD\}; \{AB, DB\};$ $\{AD, CB\}; \{AD, CD\}; \{AD, DB\}; \{CB, CD\};$ $\{CB, DB\}; \{CD, DB\}$	5	1	2	3	4	5
b	$\{AB, CB\}; \{AB, CD\}; \{AB, DB\}; \{AD, CD\};$ $\{AD, DB\}; \{CD, DB\}$	3	1	1	2	2	3
m	$\{AB, CD\}; \{AD, CD\}; \{AD, DB\}; \{CD, DB\}$	3	1	3	1	2	1
a	$\{AB, CD\}; \{AD, DB\}; \{CD, DB\}$	2	1	2	1	2	1
t	$\emptyset$	1	1	1	1	1	1

Tabela 7.1: α , A_α , X_α e $C_\alpha^{X_\alpha}$ do Exemplo 7.1

A Tabela 7.1 mostra que o número cromático de $\tilde{G}$ é $X(\tilde{G}) = \{(1, t); (2, a); (3, m); (3, b); (5, n)\}$.

Uma interpretação de $X(\tilde{G})$ pode ser a seguinte: os valores baixos de α são associados para diminuir os níveis de atenção dos condutores e, conseqüentemente, os semáforos devem controlar o tráfego, logo o número cromático é elevado; por outro lado para α com valores mais elevados os níveis de atenção dos condutores no cruzamento aumentam e o número cromático é mais baixo, permitindo que os semáforos controlem parcialmente o tráfego, ou

seja, nestes casos, por exemplo, o tráfego nas rotas AD e CD (Figura 7.2) seria liberado simultaneamente, ficando a cargo do condutor tomar cuidado ao utilizar uma destas rotas (direções).

Para resolver o problema de coloração *fuzzy* pode-se utilizar qualquer algoritmo para grafos *crisp*, G_α , que determina o número cromático. Para grafos *fuzzy* de tamanho pequeno ou médio um algoritmo exato pode ser usado, (Eppstein 2003, Herrmann e Hertz 2002, Mehrotra e Trick 1996), porém, para grafos de tamanhos maiores pode ser necessário o uso de alguma heurística (ver: Avanthay, Hertz e Zufferey (2003), di Blas, Jagota e Hughey (2003)).

7.4 O problema da (d,f) -coloração

Nesta seção é introduzida a extensão proposta por Muñoz et al. (2005). Nesta extensão as incompatibilidades *fuzzy* são determinadas através de uma medida de dissimilaridade (Definição 33) definida no conjunto de cores e de uma função escala.

É apresentado a seguir um exemplo ilustrativo para justificar este novo conceito.

7.4.1 O problema da distribuição de exames

Este problema consiste em elaborar o calendário dos exames de um conjunto de disciplinas, sendo que estes necessitam ser realizados dentro de um período de tempo prefixado e que um aluno não poderá fazer mais que um exame simultaneamente.

Uma versão simplificada deste tipo de problema pode ser modelada como um problema de coloração de grafos clássico, sendo as disciplinas os vértices do grafo, as arestas unem as disciplinas que possuem alunos matriculados em comum e as cores representam as seções em que são possíveis as realizações dos exames. Ao resolver este problema, minimiza-se o número de seções necessárias para a realização dos exames.

Contudo, em geral, nestes tipos de problemas é conveniente que os exames de certas disciplinas ocorram o mais distante possível no tempo. No entanto, utilizando o modelo clássico de coloração não é possível exigir estes tipos de restrições, já que o único que se tem

em conta ao colorir cada vértice é que a cor que se atribua ao vértice i não coincida com nenhuma das cores de seus vértices adjacentes. Apresenta-se a seguir um exemplo que ilustra o problema da (d, f) -coloração, que será descrito na subseção seguinte.

Exemplo 7.2 Distribuição de exames.

Deseja-se distribuir os exames de seis disciplinas (A, B, C, D, E, F), sendo que entre estas existem distintos graus de incompatibilidade: nulo (n), baixo (b), médio (m) e alto (a). Os elementos μ_{ij} da matriz a seguir indicam o grau de incompatibilidade entre as disciplinas i e j , $\forall i, j \in \{A, B, \dots, F\}$ tal que $i \neq j$:

$$\mu = \begin{pmatrix} - & b & a & n & a & m \\ b & - & m & n & n & n \\ a & m & - & b & n & m \\ n & n & b & - & b & m \\ a & n & n & b & - & n \\ m & n & m & m & n & - \end{pmatrix}$$

Esta situação pode ser representada mediante o seguinte grafo *fuzzy* $\tilde{G} = (V, \mu)$ (Figura 7.5), tal que $V = \{A, B, \dots, F\}$.

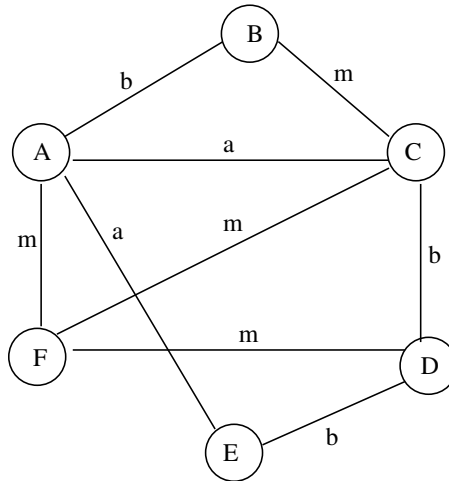


Figura 7.5: Representação do grafo $\tilde{G}$

Uma forma de interpretar o grau de incompatibilidade entre as disciplinas é a seguinte: se duas disciplinas são compatíveis (isto é, possuem incompatibilidade nula) seus exames podem ser realizados em qualquer dia, se a incompatibilidade entre as disciplinas for baixa, média ou alta, seus exames deverão ocorrer ao menos em 1, 2 ou 3 dias de distância, respectivamente.

Resolvendo este problema, através do algoritmo proposto por Muñoz *et al*, tem-se que são necessários cinco dias para a realização dos exames, sendo a seguinte distribuição ótima (Tabela 7.2):

Dias	1	2	3	4	5
Disciplinas	A,D	B	F	E	C

Tabela 7.2: Resultados da Figura 7.5

7.4.2 A coloração (d,f) -estendida de um grafo *fuzzy*

Seja $\tilde{G} = (V, \tilde{A})$ um grafo *fuzzy* com nós *crisp* e arestas *fuzzy*, onde $\tilde{A} = \{((i, j), \mu_{\tilde{A}}(i, j)) \text{ tal que } i, j \in V\}$ e $\mu_{\tilde{A}} : V \times V \rightarrow I$ é a função de pertinência (para cada aresta $(i, j) \in V \times V$, $\mu_{\tilde{A}}(i, j)$ representa o nível de intensidade de (i, j)). Sejam $\mu_{ij} = \mu_{\tilde{A}}(i, j)$, $\forall i, j \in V$ e $\mu = (\mu_{ij})_{i,j \in V}$. Dado que o conjunto de arestas $\tilde{A}$ fica perfeitamente caracterizado mediante o conjunto de vértices V e a matriz μ , logo o grafo $\tilde{G}$ pode ser denotado equivalentemente por $\tilde{G} = (V, \mu)$.

Definição 33 *Seja S o conjunto de cores possíveis, a medida de dissimilaridade d é definida por $d : S \times S \rightarrow [0, \infty)$, com as seguintes propriedades:*

1. $d(r, s) \geq 0$, $\forall r, s \in S$,
2. $d(r, s) = 0 \iff r = s$, $\forall r, s \in S$,
3. $d(r, s) = d(s, r)$, $\forall r, s \in S$.

Definição 34 *Sejam dados um grafo fuzzy $\tilde{G} = (V, \mu)$, um conjunto S de cores, uma medida de dissimilaridade d definida em S e uma função escala f . Uma função de coloração (d,f) -estendida de $\tilde{G}$, denotada por $C_{d,f}$ (para simplificar C), é uma aplicação $C : V \rightarrow S$ tal que: $d(C(i), C(j)) \geq f(\mu_{ij}) \forall i, j \in V$ tal que $i \neq j$.*

Definição 35 *Dado $k \in S$, uma k -coloração (d,f) -estendida de $\tilde{G}$ é uma função de coloração (d,f) -estendida de $\tilde{G}$ com no máximo k cores distintas.*

Uma k -coloração (d,f) -estendida de um grafo $\tilde{G} = (V, \mu)$ pode ser considerada como uma generalização de uma k -coloração de um grafo *crisp* $G = (V, A)$, fazendo $I = \{0, 1\}$, $d = d^0$ e $f(0) = 0$, $f(1) = 1$, tal que:

$$d^0(r, s) = \begin{cases} 1, & \text{se } r \neq s \\ 0, & \text{se } r = s \end{cases} \quad \forall r, s \in S \quad (7.1)$$

$\tilde{G}$ torna-se um grafo *crisp* se a matriz μ for definida da seguinte forma:

$$\mu_{ij} = \begin{cases} 1, & \text{se } (i, j) \in A \\ 0, & \text{caso contrário} \end{cases} \quad \forall i, j \in V \quad (7.2)$$

Este novo conceito apresenta algumas diferenças em relação ao caso clássico, como descrito a seguir:

Dado um grafo *crisp* $G = (V, A)$, uma função de coloração para G sempre existe, sendo $X(G) \leq |V|$. Uma função de coloração trivial é $C^{|V|} = i$, $\forall i \in V$. Entretanto, dado um grafo *fuzzy* $\tilde{G} = (V, \mu)$, uma função de coloração (d, f) -estendida para $\tilde{G}$ nem sempre existe, como ilustra o Exemplo 7.3.

Exemplo 7.3 Inexistência de solução.

Seja $\tilde{G} = (V, \mu)$ um grafo *fuzzy*, tal que

$$V = \{A, B, C\}; I = \{n, b, a\}; \mu = \begin{pmatrix} - & b & a \\ b & - & n \\ a & n & - \end{pmatrix}.$$

Seja $S = \{1, \dots, k\}$ o conjunto de cores, $d = d^0$ (Equação 7.1) e a função escala f definida na Tabela 7.3.

I	n	b	a
$f(I)$	0	1	2

Tabela 7.3: Função escala f do Exemplo 7.3

Então, uma k -coloração (d, f) -estendida não existe para qualquer $k \in \mathbb{N}$. De fato, $d(C^k(A), C^k(C)) \leq 1$ para toda k -coloração C^k e $f(\mu_{AC}) = f(a) = 2$.

Note que a condição " $d(C(i), C(j)) \geq f(\mu_{ij}) \forall i, j \in V$ tal que $i \neq j$ " não foi satisfeita neste exemplo, pois $d(C(i), C(j)) < 2 = f(\mu_{ij}) \forall i, j \in V$, logo não existe uma k -coloração (d, f) estendida para este caso.

Outra diferença com respeito a k -coloração clássica é que na k -coloração (d, f) -estendida pode haver algumas cores que não sejam atribuídas a nenhum nó, como o exemplo a seguir:

Exemplo 7.4 Ilustrativo.

Seja $\tilde{G} = (V, \mu)$ um grafo *fuzzy* (Figura 7.6),

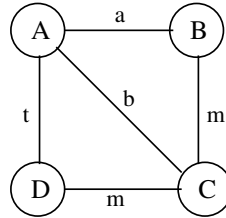


Figura 7.6: Representação do grafo $\tilde{G}$ do Exemplo 7.4

tal que: $V = \{A, B, C, D\}$; $I = \{n, b, m, a, t\}$; $\mu = \begin{pmatrix} - & a & b & t \\ a & - & m & n \\ b & m & - & m \\ t & n & m & - \end{pmatrix}$.

Seja $S = \{1, 2, 3, 4\}$ o conjunto de cores e seja d a medida de dissimilaridade definida na Tabela 7.4.

$d(r, s)$	1	2	3	4
1	0	1	2	2
2	1	0	1	1
3	2	1	0	4
4	2	1	4	0

Tabela 7.4: Medida de dissimilaridade d do Exemplo 7.4

Seja a função escala f definida na Tabela 7.5.

I	n	m	b	a	t
$f(I)$	0	1	2	3	4

Tabela 7.5: Função escala f do Exemplo 7.4

Uma 4-coloração (d, f) -estendida para este exemplo é:

$$C(A) = 3, C(B) = 4, C(C) = 1, C(D) = 4.$$

Note que a cor 2 não é associada pois, $C(i) \neq 2, \forall i \in V$.

O (d, f) -número cromático de um grafo *fuzzy* é definido, analogamente ao caso *crisp*, da seguinte forma:

Definição 36 O (d, f) -número cromático de $\tilde{G}$ é definido da seguinte forma: $X_{d,f}(\tilde{G}) = \min \{k \in S \mid \tilde{G} \text{ admite alguma } k\text{-coloração } (d, f)\text{-estendida}\}.$

Logo, o problema da (d,f) -coloração consiste em determinar o (d, f) -número cromático de um grafo *fuzzy* e uma função de coloração (d,f) -estendida associada.

Considerando no Exemplo 7.2 que o número de dias possíveis para a realização dos exames é cinco (de segunda a sexta, às 9:00) e definindo a medida de dissimilaridade como a distância $d(r, s) = |r - s|, \forall r, s \in \{1, 2, \dots, 5\}$ e a função f como $f(n) = 0, f(b) = 1, f(m) = 2$ e $f(a) = 3$, este problema se reduz a resolver o problema da (d, f) -coloração para $\tilde{G}$.

O (d, f) -número cromático encontrado é $X_{d,f}(\tilde{G}) = 5$ e uma 5-coloração (d, f) -estendida de $\tilde{G}$ é definida da seguinte forma:

$$C(A) = 1, C(B) = 2, C(C) = 5, C(D) = 1, C(E) = 4, C(F) = 3.$$

Portanto, o número mínimo de dias necessários para realizar os exames é 5, e uma distribuição ótima é a seguinte:

Dias	Seg	Ter	Qua	Qui	Sex
Exames	A,D	B	F	E	C

Tabela 7.6: Resultados do Exemplo 7.2

7.5 O problema da (d,f,g) -coloração

Nesta seção é apresentada uma extensão da (d, f) -coloração (ou coloração (d, f) -estendida), que será denominada (d,f,g) -coloração (ou coloração (d, f, g) -estendida). Para introduzir este novo conceito é apresentado um exemplo ilustrativo.

7.5.1 O problema da alocação de horários de disciplinas

Este problema consiste em elaborar os horários de um conjunto de disciplinas, porém, para a elaboração destes existem algumas restrições: há um número máximo de disciplinas por dia e existem graus de incompatibilidade entre as mesmas, isto é, disciplinas consideradas "difíceis" deverão ter uma certa distância de dias entre elas, ou seja, disciplinas com grau de incompatibilidade alto não poderão ser ministradas consecutivamente.

Exemplo 7.5 Alocação de horários.

O coordenador do curso de Ciência da Computação de uma determinada universidade necessita distribuir os horários das disciplinas do seu curso, porém ele sabe que cada disciplina possui 4h/a semanais, sendo ministradas em somente um dia, também sabe-se que os alunos não podem ter mais do que duas disciplinas por dia (8h/a) e que as mesmas deverão ser ministradas somente em dias úteis. Além das restrições supracitadas, algumas disciplinas são incompatíveis, isto deve-se ao grau de dificuldade das mesmas, logo, disciplinas consideradas difíceis não devem ser alocadas no mesmo dia. Os graus de incompatibilidade das mesmas são: nulo, baixo e médio, isto significa que estas devem ser ministradas com uma diferença de zero, um ou dois dias.

Dado V o conjunto das sete disciplinas e μ_{ij} a matriz que indica o grau de incompatibilidade entre as disciplinas i e j , $\forall i, j \in \{1, 2, \dots, 7\}$ tal que $i \neq j$, como fica a distribuição das disciplinas e qual é o número mínimo de dias que estas disciplinas são alocadas?

Seja:

$V = \{\text{álgebra linear (AL)}, \text{cálculo diferencial (CD)}, \text{física (F)}, \text{programação de computadores (PC)}, \text{matemática discreta (MD)}, \text{organização de computadores (OC)}, \text{introdução à computação (IC)}\}$; e

$$\mu = \begin{pmatrix} - & m & b & n & m & n & n \\ m & - & b & b & b & n & n \\ b & b & - & b & m & n & n \\ n & b & b & - & b & b & n \\ m & b & m & b & - & m & n \\ n & n & n & b & m & - & b \\ n & n & n & n & n & b & - \end{pmatrix}$$

Esta situação pode ser representada pelo grafo da Figura 7.7.

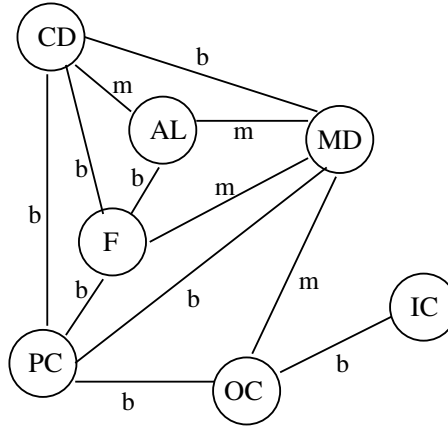


Figura 7.7: Grafo $\tilde{G}$ do Exemplo 7.5

Resolvendo este problema, através do algoritmo proposto para a (d, f, g) -coloração (Seção 7.6.2), tem-se que são necessários quatro dias, sendo a seguinte distribuição:

Dias	1	2	3	4
Disciplinas	AL, PC	F, OC	IC, CD	MD

Tabela 7.7: Resultados da Figura 7.7

7.5.2 A coloração (d,f,g) -estendida de um grafo *fuzzy*

A função de coloração (d, f, g) -estendida é uma extensão da (d, f) -estendida, sendo que esta possui a restrição de que uma determinada cor não poderá ser atribuída mais do que g vezes. Portanto as definições a seguir são baseadas nas da função (d, f) -estendida, acrescidas da restrição do número máximo de vezes que cada cor pode ser atribuída.

Definição 37 Dado um grafo fuzzy $\tilde{G} = (V, \mu)$, um conjunto S de cores, uma medida de dissimilaridade d definida em S , uma função escala f e um valor g representando o número máximo de vezes que cada cor poderá ser atribuída, então uma função de coloração (d, f, g) -

estendida de $\tilde{G}$, denotada por $C_{d,f,g}$ (para simplificar C), é uma aplicação $C : V \rightarrow S$ tal que:

$$d(C(i), C(j)) \geq f(\mu_{ij}) \quad \forall i, j \in V \text{ sendo } i \neq j; \text{ e}$$

$$\text{card}(A_j) \leq g, \quad \forall j = \{1, 2, \dots, k\}, \text{ sendo } A_j = \{i \in V \text{ tal que } C(i) = s_j; s_j \in S\}$$

Definição 38 Dado $k \in S$, uma k -coloração (d,f,g) -estendida de $\tilde{G}$ é uma função de coloração (d,f,g) -estendida de $\tilde{G}$ com no máximo k cores distintas.

Definição 39 O (d, f, g) -número cromático de $\tilde{G}$ é definido da seguinte forma: $X_{d,f,g}(\tilde{G}) = \min \{k \in S \mid \tilde{G} \text{ admite alguma } k\text{-coloração } (d, f, g)\text{-estendida}\}.$

Logo, o problema da (d,f,g) -coloração consiste em determinar o (d, f, g) -número cromático de um grafo fuzzy e uma função de coloração (d,f,g) -estendida associada.

Considerando, no Exemplo 7.5, que os dias possíveis para a distribuição dos horários das aulas sejam de segunda a sexta, definindo a medida de dissimilaridade como a distância $d(r, s) = |r - s|$, $\forall r, s \in \{1, 2, \dots, 5\}$, atribuindo à função f os valores $f(n) = 0$, $f(b) = 1$ e $f(m) = 2$ e considerando que poderão ser ministradas no máximo duas disciplinas por dia, este problema se reduz a resolver o problema da (d, f, g) -coloração para $\tilde{G}$.

Logo, o (d, f, g) -número cromático deste exemplo é $X_{d,f,g}(\tilde{G}) = 4$ e a distribuição ótima das aulas fica da seguinte forma (Tabela 7.8):

Dias	Seg	Ter	Qua	Qui
Disciplinas	AL,PC	F,OC	IC,CD	MD

Tabela 7.8: Resultado do Exemplo 7.5.

7.6 Algoritmos propostos

Nesta seção são abordados os algoritmos propostos para as extensões (d, f) e (d, f, g) .

7.6.1 Algoritmo proposto por Muñoz et al para a (d,f) -coloração

Em Muñoz et al. (2005) foi proposto um algoritmo exato para resolver o problema da (d,f) -coloração. Tal algoritmo está baseado no algoritmo proposto por Brown (1972) para obter o

número cromático de um grafo *crisp*.

Este algoritmo é composto de duas subrotinas, *colvoraz* e (d, f) -coloração, sendo a segunda o programa principal. A subrotina *colvoraz*, que é chamada diversas vezes, tem por finalidade colorir sucessivamente os vértices $v_0, v_1, \dots, v_n$.

Algoritmo proposto por Muñoz et al

Dado um vértice v_0 , a subrotina *colvoraz* colore sucessivamente os vértices $v_0, v_1, \dots, v_n$, aplicando um procedimento que atribui aos vértices a menor cor possível, de forma que resulte uma função de coloração (d, f) -estendida de $\tilde{G}$ (supõe-se que os vértices $1, 2, \dots, v_0 - 1$ já estão coloridos). Os parâmetros desta subrotina são os seguintes:

- v_0 : primeiro vértice a colorir;
- C : função de coloração (d, f) -estendida;
- c_{max} : maior cor atribuída até o momento;
- v_{max} : primeiro vértice que possui a cor c_{max} ; e
- k : número de cores disponíveis.

O algoritmo (d, f) -coloração resolve este problema realizando diversas chamadas à subrotina *colvoraz*. Para determinar se existe alguma k -coloração (d, f) -estendida de $\tilde{G}$, este algoritmo começa chamando a subrotina *colvoraz*, tomando $v_0 = 1$ e $c_{max} = k + 1$. Se esta subrotina devolver $c_{max} \leq k$, então existe alguma k -coloração (d, f) -estendida de $\tilde{G}$ e c_{max} é um limitante superior para o (d, f) -número cromático. Dada a coloração atual C e seus parâmetros associados c_{max} e v_{max} , o procedimento de melhora do algoritmo busca um vértice $v_m < v_{max}$ tal que sua cor possa ser incrementada sem que seja necessário utilizar uma cor maior que c_{max} . Inicialmente se toma $v_m = v_{max} - 1$, e a cada iteração o valor de v_m vai diminuindo em uma unidade até $v_m = 1$; cada vez que o valor devolvido de c_{max} melhora o limitante superior atual para o número cromático, atualiza-se o determinado limitante.

Alterações no algoritmo proposto

Analisando o algoritmo proposto nota-se que em momento algum é comentado como atribuir o valor inicial de k , ou ao menos como determinar um valor inicial para k . Em pequenas aplicações é possível associar o conjunto das cores e, conseqüentemente, o valor de k . Já para aplicações de grande porte é difícil atribuir um valor inicial para k . Como esta variável (k) é o limitante superior inicial e, à medida em que ocorrem as iterações, o ideal é que seu valor diminua, propõe-se que seja atribuído como valor inicial o produto entre o número de nós da rede (n) com a maior distância (função escala definida) entre todos os arcos da rede (sendo que, neste caso, as cores estão relacionadas a números).

7.6.2 Algoritmo proposto para o problema da (d,f,g)-coloração

O algoritmo proposto nesta seção é baseado no algoritmo de Muñoz et al. (2005) para o problema da (d, f) -coloração, possuindo algumas alterações para tratar da restrição do número máximo de vezes que cada cor pode assumir, além de possuir o cálculo do valor inicial de k .

Este algoritmo também possui duas subrotinas, sendo a primeira o programa principal e a segunda, chamada *colorir*, que tem por finalidade colorir sucessivamente os vértices $v_0, v_1, \dots, v_n$.

O algoritmo (d, f, g) -coloração resolve o problema da (d, f, g) -coloração realizando diversas chamadas à subrotina *colorir*. O funcionamento deste é semelhante ao algoritmo da (d, f) -coloração, cuja descrição foi feita na subseção anterior.

A seguir é apresentado o referido algoritmo e a subrotina *colorir*.

Informações sobre o algoritmo proposto

C : função de coloração (d, f, g) -estendida;

k : maior cor que poderá ser atribuída;

v_0 : primeiro vértice a colorir;

w : quantidade de vezes que a cor atribuída ao nó i ($C(i)$) já foi atribuída;

c_{max} : maior cor associada aos vértices; e

v_{max} : primeiro vértice que recebeu a cor c_{max} .

(d,f,g)-coloração

PASSO 1: Inicialização das variáveis

1. $C(1)=0$ atribuição da cor inicial ao nó 1
2. Entre com os dados da rede que será colorida
3. Forneça o valor de g
4. $k = n * (\text{maior} - \text{distancia})$ cálculo do valor inicial de k
5. $c_{max} = k + 1$
6. Pare=falso

PASSO 2: Coloração do nó 1.

- Colorir(1,C, v_{max} , c_{max})

PASSO 3: Coloração dos demais nós.

1. Se ($c_{max} \leq k$) faça
 - Para $i = 1, \dots, n \Rightarrow col(i) = C(i)$
 - Enquanto (Pare=falso) faça
 - (a) Se $w \leq g$ (verificação da restrição do número de cores)
 - $col(i) = C(i)$
 - Pare=verdadeiro
 - (b) Senão $col(i) = C(i) + 1$
 - $ncrom = c_{max}$
 - $v_m = v_{max} - 1$
 - Pare=falso
 - Enquanto ($v_m \geq 1$) faça
 - (a) Se $C(v_m) < ncrom - 1$
 - i. $C(v_m) = C(v_m) + 1$
 - ii. Colorir(v_m, C, c_{max}, v_{max})
 - iii. Se ($c_{max} < ncrom$)
 - Para $i = 1, \dots, n \Rightarrow col(i) = C(i)$

- Enquanto (Pare=falso) faça
 - A. Se $w \leq g$ (verificação da restrição do número de cores)
 - * $col(i) = C(i)$
 - * Pare=verdadeiro
 - B. Senão $col(i) = C(i) + 1$
- $ncrom = c_{max}$
- $v_m = v_{max} - 1$

iv. Senão $\Rightarrow v_m = v_m - 1$

(b) Senão $\Rightarrow v_m = v_m - 1$

- Imprima(col é uma ncrom-coloração (d,f,g)-estendida)

2. Imprima(não existe nenhuma k-coloração (d,f,g)-estendida)

A subrotina utilizada para colorir os nós da rede é a seguinte:

(d,f,g)-coloração

PASSO 1: Inicialização das variáveis

1. Pare=0
2. $i = v_0$ (nó que será colorido)
3. $c = C(i) + 1$ (cor inicialmente atribuída ao nó i)

PASSO 2: Atribuição de cor ao nó i .

1. Enquanto (Pare=falso) faça (Colorir o nó v_0)
 - $j = 1$
 - Enquanto ($j < i$) faça
 - (a) Se $(d(c, C(j)) < f(\mu_{ij}))$ (nós i e j não são compatíveis)
 - $c = c + 1$
 - $j = 1$
 - Se $(c \geq c_{max}) \Rightarrow j = i$ (c_{max} : maior cor atribuída)

- Senão $j = j + 1$ (i e j são compatíveis)
- Se $c < c_{max}$ (se a cor atribuída ao nó i for menor que c_{max} atribui-se a referida cor ao nó i)
 - $C(i)=c$
 - Enquanto (Pare=falso) faça
 - (a) Se $w \leq g \Rightarrow C(i) = c$ (verificação da restrição do número de cores)
 - (b) Senão $C(i) = c + 1$ ($C(i)=c$ é compatível)
 - $i = i + 1$
 - $c = 1$
 - Se $(i > n) \Rightarrow \text{Pare}=\text{verdadeiro}$
- Senão
 - Se $i \leq v_0 \Rightarrow \text{Pare}=\text{verdadeiro}$
 - Senão
 - (a) Enquanto $((i > v_0 \text{ e } C(i - 1) \geq c_{max} - 1)) \Rightarrow i = i - 1$
 - (b) $i = i - 1$
 - (c) $c = C(i) + 1$
 - (d) Se $i = 0 \Rightarrow c = 1$
 - (e) Senão $c = C(i) + 1$
 - (f) Se $(i < v_0) \Rightarrow \text{Pare}=\text{verdadeiro}$

PASSO 3: Atribuição final da cor ao nó i .

- Se $(i > n)$
 1. $c_{max} = 0$
 2. Para $i = 1, \dots, n$
 - (a) Se $(C(i) > c_{max})$
 - i. $c_{max} = C(i)$
 - ii. $v_{max} = i$

PASSO 4: Retorne ao programa principal.

7.7 Resultados computacionais

Nesta seção são comentados apenas alguns resultados, visto que as soluções para cada um dos exemplos citados nas seções anteriores foram encontradas aplicando os respectivos algoritmos.

Exemplo 7.6 Alocação de horários.

Aplicando o algoritmo da (d, f) -coloração, no Exemplo 7.5 (Figura 7.7) tem-se que o (d, f) -número cromático também é $X_{d,f}(\tilde{G}) = 4$, semelhante ao valor de (d, f, g) -coloração, porém a distribuição das aulas fica da seguinte forma:

Dias	Seg	Ter	Qua	Qui
Disciplinas	AL, PC, IC	F, OC	CD	MD

Tabela 7.9: Resultado do Exemplo 7.6 usando (d, f) -coloração

Analisando os resultados das duas extensões, (d, f) -estendida e (d, f, g) -estendida, (Tabelas 7.8 e 7.9) percebe-se que a restrição do número máximo de vezes atribuído a cada cor resultou numa distribuição diferente das disciplinas.

Exemplo 7.7 Ilustração.

Seja o grafo $\tilde{G} = (V, \mu)$ (Figura 7.8) um grafo *fuzzy*,

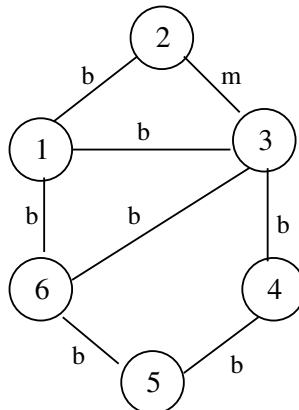


Figura 7.8: Grafo de incompatibilidades do Exemplo 7.7

$$\text{tal que: } V = \{1, 2, 3, 4, 5, 6\}; I = \{n, b, m\}; \mu = \begin{pmatrix} - & b & b & n & n & b \\ b & - & m & n & n & n \\ b & m & - & b & n & b \\ n & n & b & - & b & n \\ n & n & n & b & - & b \\ b & n & b & n & b & - \end{pmatrix}$$

Dada a medida de dissimilaridade d como a distância $d(r, s) = |r - s|$, $\forall r, s \in \{1, 2, \dots, k\}$, a função escala f como $f(n) = 0$, $f(b) = 1$ e $f(m) = 2$ e $g = 2$ (número máximo de nós que pode receber a mesma cor). Aplicando as funções de coloração (d, f) -estendida e (d, f, g) -estendida tem-se as seguintes soluções:

- O (d, f) -número cromático encontrado é $X_{d,f}(\tilde{G}) = 3$ e uma 3-coloração (d, f) -estendida de $\tilde{G}$ é definida da seguinte forma:

$$C(1) = 2, C(2) = 1, C(3) = 3, C(4) = 1, C(5) = 2, C(6) = 1.$$

- O (d, f, g) -número cromático encontrado é $X_{d,f,g}(\tilde{G}) = 4$ e uma 4-coloração (d, f) -estendida de $\tilde{G}$ é definida da seguinte forma:

$$C(1) = 1, C(2) = 2, C(3) = 4, C(4) = 1, C(5) = 2, C(6) = 3.$$

Portanto, neste exemplo verifica-se que a restrição referente ao número máximo de vezes que cada cor pode receber alterou o resultado do número cromático.

7.8 Considerações finais

Neste capítulo foi abordado o conceito de coloração de grafos *fuzzy*. Inicialmente foi apresentada a extensão do conceito de incompatibilidade entre os nós de um grafo, para assim incluir o conceito de incertezas e imprecisões (Muñoz et al. 2005). Para esta nova situação, foram detalhadas duas extensões, propostas por Muñoz et al (Muñoz et al. (2003), Muñoz et al. (2005)), para o problema de coloração de grafos *fuzzy* com nós *crisp* e arestas *fuzzy*. A primeira é baseada na fuzzificação natural do problema clássico; esta foi exemplificada com o problema dos semáforos, sendo que seu número cromático pode ser definido e interpretado de

acordo com o significado das arestas do grafo *fuzzy*. A segunda proposta estende o conceito da função de coloração *crisp* permitindo a separação de nós coloridos através da medida de dissimilaridade, definida sobre o conjunto de cores e sobre a função escala. Esta extensão é chamada coloração (d, f) -estendida. O novo conceito de coloração considera um grau de incompatibilidade relacionado ao valor da função de pertinência de cada aresta. A introdução da função escala permite ao usuário um parâmetro de controle de ordem para obter as soluções válidas que verificam as restrições “*as arestas mais incompatíveis*”, “*as cores mais distantes*”. Um algoritmo exato para este problema de coloração estendido foi introduzido e foram apresentados alguns resultados computacionais.

Com base nas extensões abordadas por Muñoz et al, neste capítulo foi introduzido uma terceira extensão para este problema, chamada coloração (d, f, g) -estendida. Esta extensão considera a separação dos nós coloridos através da medida de dissimilaridade, semelhante à (d, f) -coloração, porém, também possui a restrição de que uma determinada cor não pode ser atribuída mais do que g -vezes. Este problema foi exemplificado (Seção 7.5.1) pelo problema de alocações de horários de disciplinas. Também foi apresentado um algoritmo exato para a solução desta extensão.

Capítulo 8

Conclusões e Trabalhos Futuros

Neste capítulo são apresentadas as conclusões da tese, as perspectivas de trabalhos futuros e os trabalhos publicados.

8.1 Conclusões

Nesta tese foram estudados alguns tipos de incertezas em problemas clássicos de grafos e fluxo em redes, do ponto de vista da teoria e dos algoritmos associados. Os problemas abordados foram: problema de caminho mínimo com incertezas (Capítulo 4), problema de fluxo máximo com incertezas (Capítulo 5), problema de fluxo de custo mínimo com incertezas (Capítulo 6) e o problema de coloração de grafos *fuzzy* (Capítulo 7). Para estes problemas foram propostos alguns algoritmos que contornam desvantagens detectadas nos trabalhos da literatura. Além destes algoritmos propostos, também foi apresentada uma definição adicional para o problema de coloração de grafos *fuzzy*.

No Capítulo 4 foi abordado o problema de caminho mínimo em grafos com estrutura *crisp* e parâmetros incertos (*fuzzy*). Neste capítulo foi feita uma minuciosa revisão bibliográfica do tema, sendo detectado que os trabalhos da literatura são aplicáveis somente em redes com parâmetros (custos) não-negativos. Com base nisto, foi proposto um algoritmo que contorna esta desvantagem, sendo este uma adaptação do algoritmo clássico de Ford-Moore-Bellman. Além da vantagem supracitada, este possui uma boa complexidade computacional quando comparado com os demais da literatura. Verificando também a inexistência de algoritmos para grafos com restrições de tempo, o algoritmo proposto foi adaptado para este tipo de

problema, sendo que cada arco possui um custo e um tempo *fuzzy*. Também foi adaptada, ao mesmo algoritmo, a heurística e o algoritmo, propostos por Chuang e Kung (2005), para encontrar o custo mínimo *fuzzy* do grafo. E, finalmente, ao mesmo algoritmo foi adaptada uma relação de ordem genérica para encontrar a solução final, ficando a cargo do usuário propor ou escolher uma relação de ordem da literatura que melhor adapte ao seu problema.

O problema de fluxo máximo em redes com estrutura *crisp* e parâmetros incertos foi abordado no Capítulo 5. Além da revisão bibliográfica do tema, neste capítulo foi apresentado um algoritmo que contorna a principal desvantagem dos algoritmos da literatura, que é a exigência que o usuário forneça o fluxo que deseja trafegar na rede e a partir daí é verificada com que pertinência este fluxo pode ser transitado. Já o algoritmo proposto apresenta o fluxo máximo e sua respectiva pertinência, não exigindo do usuário o fluxo desejável, o que é uma grande vantagem quando se trabalha com redes de grande porte.

O problema de fluxo de custo mínimo para redes *crisp* com parâmetros *fuzzy* foi apresentado no Capítulo 6, onde além da revisão bibliográfica do tema foram propostos alguns algoritmos. Inicialmente verificou-se que os principais trabalhos da literatura utilizam índices de defuzzificação para transformar o problema *fuzzy* em um problema *crisp* e, a partir de então, utilizam um algoritmo clássico para resolver tal problema. Com base nisto, neste capítulo foram propostos tanto um algoritmo que transforma o problema incerto em um *crisp*, quanto algoritmos que trabalham com o problema incerto. Para o primeiro caso foi utilizado o algoritmo clássico do Big-M, este algoritmo foi proposto para casos em que o usuário deseja obter uma solução aproximada e rápida, visto que é um algoritmo de fácil implementação. Já para o segundo caso (trabalhar com as incertezas) foram propostos quatro algoritmos, sendo um para os casos em que a rede possui somente um nó origem e um nó destino, o segundo para o caso da rede possuir uma origem e múltiplos nós destinos, o terceiro para o caso da rede ter múltiplos nós origens e um destino e finalmente o último para o caso de múltiplas origens e múltiplos destinos. Pode-se dizer que estes quatro últimos algoritmos são inéditos, visto que não se tem conhecimento de nenhum trabalho deste tema que mantém o problema incerto até o final.

Problema envolvendo grafo com estrutura incerta (*grafo fuzzy*) foi abordado no Capítulo 7, que trata do problema da coloração de grafos. Neste capítulo foram apresentados os trabalhos da literatura que abordam este tema (Ramírez 2001, Muñoz et al. 2003, Muñoz et al. 2005) e foi proposta uma nova definição para o problema de coloração *fuzzy*, chamada (d, f, g) -coloração (ou coloração (d, f, g) -estendida), cujo objetivo é de tratar o problema de coloração com algumas restrições. Também foi proposto um algoritmo para tal definição, sendo este uma adaptação dos algoritmos de Muñoz et al. (2005).

Portanto, a principal contribuição desta tese foi apresentar algoritmos para suprir desvantagens ou vazios da literatura sobre os temas abordados.

No âmbito pessoal, pode-se dizer que este período de doutorado foi muito gratificante, visto que tivemos contato com excelentes pesquisadores da Unicamp e do estrangeiro (durante o período de estágio de doutoramento na Universidade de Granada, Espanha).

8.2 Trabalhos futuros

- Problema de caminho mínimo:
 - algoritmo com restrição de tempo (Seção 4.5): estudar como tratar quando há impasse na escolha do melhor caminho, pois ocorrerão casos em que o caminho mais barato satisfaz as restrições de tempo com baixa pertinência, enquanto que um caminho mais caro satisfaz as restrições de tempo com uma pertinência maior;
 - algoritmo com a heurística de Chuang e Kung (Seção 4.6): estudar detalhadamente o comportamento deste algoritmo em redes com diversos tamanhos;
 - algoritmo com relação de ordem genérica (Seção 4.7): implementar outras relações de ordem;
 - implementar um sistema de suporte a decisão.
- Problema de fluxo máximo:

- estudar outras formas de tratar as incertezas das capacidades;
 - trabalhar com redes cujo limitante inferior das capacidades seja diferente de zero;
 - estudar a viabilidade de adaptar este algoritmo ao proposto em redes com estruturas incertas.
- Problema de fluxo de custo mínimo: estudar novos índices de comparação entre números *fuzzy* para efeito de comparação com os adotados.
 - Coloração de grafos: estudar a viabilidade de algoritmos aproximados para este problema, visto que o proposto poderá ser lento quando aplicado em redes de grande porte.

Outros problemas que podem ser estudados utilizando grafos e a teoria de conjuntos *fuzzy*:

- problemas de transporte;
- problemas *multicommodity*.

8.3 Trabalhos publicados

Durante este período foram publicados e submetidos os seguintes trabalhos:

- (Hernandes e Yamakami 2004) “Um algoritmo para o problema de caminho mínimo em grafos com custos nos arcos *fuzzy*”, apresentado no XV Congresso Brasileiro de Automática;
- (Hernandes e Takahashi 2004) “Adaptação de um algoritmo de caminho mínimo em grafos com custos nos arcos e restrições de tempo *fuzzy*”, apresentado no XXXVI Simpósio Brasileiro de Pesquisa Operacional;
- (Hernandes et al. 2005) “Uso da similaridade entre caminhos para o problema de caminho mínimo com parâmetros *fuzzy*”, apresentado no XXXVII Simpósio Brasileiro de Pesquisa Operacional;

- (Hernandes, Yamakami, Takahashi e Verdegay 2006) “Um algoritmo para o problema de fluxo máximo em redes com incertezas”, apresentado no XXXVIII Simpósio Brasileiro de Pesquisa Operacional;
- (Hernandes, Lamata, Takahashi, Verdegay e Yamakami 2006) “Uma proposta de solução para o problema de caminho mínimo em grafos com parâmetros incertos”, apresentado no XVI Congresso Brasileiro de Automática;
- (Hernandes, Lamata, Verdegay e Yamakami 2006) “A generic algorithm for the shortest path problem on a network with *fuzzy* parameters”, apresentado no International Symposium on *Fuzzy* and Rough Sets;
- “On the solution of *fuzzy* shortest path problem”, apresentado no XIII Congreso Español sobre Tecnologías y Lógica *Fuzzy* (2006). Trabalho apresentado, porém os anais do congresso ainda não foram enviados; e
- “The shortest path problem on networks with *fuzzy* parameters”, submetido, em junho de 2006, à revista *Fuzzy Sets and Systems* (artigo aceito, *in press*).

Referências Bibliográficas

- Ahuja, R. K., Magnanti, T. L. e Orlin, J. B. (1993). *Network Flows*, Prentice Hall.
- Ali, M., Ramamurthy, B. e Deogun, J. S. (2003). A genetic algorithm for routing in wdm optical networks with power considerations. part 1: The unicast case, <http://hamsa.unl.edu/byra/professional>.
- Avanthay, C., Hertz, A. e Zufferey, N. (2003). A variable neighborhood search for graph coloring, *European Journal of Operational Research* **151**(2): 131–141.
- Bazarra, M., Jarvis, J. e Sherali, H. F. (1990). *Linear Programming and Network Flows*, John Wiley.
- Bellman, R. E. (1958). On a routing problem, *Quarterly Applied Mathematics* (16): 87–90.
- Bellman, R. E. e Zadeh, L. (1970). Decision making in a fuzzy environment, *Management Science* **17**(4): B:141–B:164.
- Berthold, M. e Hand, D. J. (1999). *Intelligent Data Analysis: An introduction*, Springer-Verlag.
- Blue, M., Bush, B. e Puckett, J. (2002). Unified approach to fuzzy graph problems, *Fuzzy Sets and Systems* (125): 355–368.
- Brown, J. I. (1972). Chromatic scheduling and the chromatic number problem, *Management Science* **19**: 456–463.
- 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 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.
- Chang, P. e Lee, E. (1999). Fuzzy decision networks and deconvolution, *Computers and Mathematics with Applications* **37**(11–12): 53–63.
- Chartrand, G. e Oellermann, O. R. (1993). *Applied and Algorithmic Graph Theory*, McGraw-Hill.
- Cheng, C.-H. (1998). A new approach for ranking fuzzy numbers by distance method, *Fuzzy Sets and Systems* **95**: 307–317.
- Chuang, T.-N. e Kung, J.-Y. (2005). The fuzzy shortest path length and the corresponding shortest path in a network, *Computers and Operations Research* **32**: 1409–1428.
- Delgado, M. R. B. S. (2002). *Projeto Automático de Sistemas Nebulosos: Uma Abordagem Co-Evolutiva*, PhD thesis, Faculdade de Engenharia Elétrica e de Computação/UNICAMP.
- Delgado, M., Verdegay, J. L. e Vila, M. A. (1985). On fuzzy tree definition, *European Journal of Operational Research* **22**: 243–249.
- Delgado, M., Verdegay, J. L. e Vila, M. A. (1988). A procedure for ranking fuzzy numbers using fuzzy relations, *Fuzzy Sets and Systems* **26**: 49–62.
- Delgado, M., Verdegay, J. L. e Vila, M. A. (1990). On valuation and optimization problems in fuzzy graphs: A general approach and some particular cases, *ORSA Journal on Computing* **2**: 74–84.
- di Blas, A., Jagota, A. e Hughey, R. (2003). A range-compact heuristic for graph coloring, *Journal of Heuristics* **9**(6): 379–388.
- Dubois, D. e Prade, H. (1980). *Fuzzy Sets and Systems: Teory and Applications*, Academic Press - New York.
- Dubois, D. e Prade, H. (1983). Ranking fuzzy numbers in the setting of possibility theory, *Information Sciences* **30**: 183–224.
- Eppstein, D. (1994). Finding the k-shortet paths, *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pp. 154–165.

- Eppstein, D. (2003). Small maximal independent sets and faster exact graph coloring, *Journal of Graph Algorithms and Applications* **7**(2): 131–140.
- García, M. S. e Lamata, M. T. (2005). The fuzzy sets in maintenance process, *Proceedings of the European Society for Fuzzy Logic and Technology*, pp. 118–123.
- Goldberg, M. C. e Luna, H. P. L. (2000). *Otimização Combinatória e Programação Linear*, Editora Campus: Rio de Janeiro.
- Gondran, M. e Minoux, M. (1984). *Graphs and Algorithms*, John Wiley and Sons, New York.
- Hernandes, F., Lamata, M. T., Takahashi, M. T., Verdegay, J. L. e Yamakami, A. (2006). Uma proposta de solução para o problema de caminho mínimo em grafos com parâmetros incertos, *XVI Congresso Brasileiro de Automática*, Salvador, BA.
- Hernandes, F., Lamata, M. T., Verdegay, J. L. e Yamakami, A. (2006). A generic algorithm for the shortest path problem on a network with fuzzy parameters, *International Symposium on Fuzzy and Rough Sets*, Santa Clara, Cuba.
- Hernandes, F. e Takahashi, M. T. (2004). Adaptação de um algoritmo de caminho mínimo em grafos com custos nos arcos e restrições de tempo fuzzy, *XXXVI Simpósio Brasileiro de Pesquisa Operacional*, São João Del Rei, MG, pp. 2227–2238.
- Hernandes, F. e Yamakami, A. (2004). Um algoritmo para o problema de caminho mínimo em grafos com custos nos arcos fuzzy, *XV Congresso Brasileiro de Automática*, Gramado, RS.
- Hernandes, F., Yamakami, A. e Takahashi, M. T. (2005). Uso da similaridade entre caminhos para o problema de caminho mínimo com parâmetros fuzzy, *XXXVII Simpósio Brasileiro de Pesquisa Operacional*, Gramado, RS, pp. 2382–2393.
- Hernandes, F., Yamakami, A., Takahashi, M. T. e Verdegay, J. L. (2006). Um algoritmo para o problema de fluxo máximo em redes com incertezas, *XXXVIII Simpósio Brasileiro de Pesquisa Operacional*, Goiânia, GO, pp. 2272–2283.
- Herrmann, F. e Hertz, A. (2002). Finding the chromatic number by means of critical graphs, *Journal of Experimental Algorithmics* **7**: 1–9.
- 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.

- Ishibuchi, H. e Tanaka, H. (1990). Multiobjective programming in optimization of the interval objective function, *European Journal of Operational Research* **48**: 219–225.
- Kerre, E. E. (1993). Basic principles of fuzzy set theory for the representation and manipulation of imprecision and uncertainty, *Communication and Cognition* pp. 57–70.
- Kim, K. e Roush, F. (1982). Fuzzy flows on networks, *Fuzzy Sets and Systems* **8**: 35–38.
- Klein, C. M. (1991). Fuzzy shortest paths, *Fuzzy Sets and Systems* **39**: 27–41.
- Kuchta, D. (2002). A generalisation of an algorithm solving the fuzzy multiple choice knapsack problem, *Fuzzy Sets and Systems* (127): 131–140.
- Lawler, E. L. (1976). *Combinatorial Optimization: Networks and Matroids*, New York: Holt, Rinehart and Winston.
- Lin, C. e Chern, M. S. (1993). The fuzzy shortest path problem and its most vital arcs, *Fuzzy Sets and Systems* (58): 343–353.
- Liou, T.-S. e Wang, M.-J. (1992). Ranking fuzzy numbers with integral interval, *Fuzzy Sets and Systems* (50): 247–255.
- Liu, S.-T. e Kao, C. (2004). Network flow problems with fuzzy arc lengths, *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics* (34): 765–769.
- Malik, D. e Moderson, J. N. (2001). *Fuzzy Discrete Structures*, Physica-Verlag, New York.
- Mehrotra, A. e Trick, M. A. (1996). A column generation approach for graph coloring, *INFORMS Journal on Computing* **8**(4): 344–354.
- Muñoz, S., Ortuño, M. T., Ramírez, J. e Yáñez, J. (2005). Coloring fuzzy graphs, *Omega* **33**: 211–221.
- Muñoz, S., Ortuño, M. T., Ramírez, J. e Yáñez, J. (2003). Coloración de grafos difusos: Aplicación a problemas de planificación de exámenes, *Anais do 27 Congresso Nacional de Estatística e Investigação Operativa (Em Espanhol)*.
- Nayeem, S. M. A. e Pal, M. (2005). Shortest path problem on a network with imprecise edge weight, *Fuzzy Optimization and Decision Making* **4**: 293–312.
- Okada, S. (2004). Fuzzy shortest path problems incorporating interactivity among paths, *Fuzzy Sets and Systems* **142**(3): 335–357.

- 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*, London: MIT Press.
- Philips, D. T. e Garcia-Diaz, A. (1981). *Fundamentals of Network Analysis*, Englewood Cliffs, NJ: Prentice-Hall.
- Ramírez, J. (2001). Extensiones del problema de coloración de grafos, *Tese de doutorado. Universidad Complutense de Madrid*.
- Rosenfeld, A. (1975). *Fuzzy Graphs, Fuzzy Sets and Their Applications*, Academic Press.
- Sengupta, A. e Pal, T. K. (2000). On comparing interval numbers, *European Journal of Operational Research* (127): 29–43.
- Shih, H.-S. e Lee, E. S. (1999). Fuzzy multi-level minimum cost flow problems, *Fuzzy Sets and Systems* **107**: 159–176.
- Takahashi, M. T. (2004). *Contribuições ao Estudo de Grafos Fuzzy: Teoria e Algoritmos*, PhD thesis, Faculdade de Engenharia Elétrica e de Computação/UNICAMP.
- Tan, L. G. e Sinclair, M. C. (1995). Wavelength assignment between the central nodes of the cost239 european optical network, *11th UK Performance Engineering Workshop*, Liverpool, pp. 235–247.
- Tan, S., Yu, Y. e Wang, P.-Z. (1998). Building fuzzy graphs from samples of nonlinear functions, *Fuzzy Sets and Systems* **93**: 337–352.
- Yager, R. R. (1978). Ranking fuzzy subsets over the unit interval, *Proceedings of the CDC*, pp. 1435–1437.
- Yager, R. R. (1980). On choosing between fuzzy subsets, *Kybernets* (9): 151–154.
- Yager, R. R. (1981). A procedure for ordering fuzzy subsets of the unit interval, *Informations Sciences* (24): 143–161.
- Zadeh, L. (1965). Fuzzy sets, *Journal of Information and Control* (8): 338–353.
- Zadeh, L. (1968). Fuzzy algorithms, *Journal of Information and Control* (12): 94–102.

Zadeh, L. (1978). Fuzzy sets as a theory of possibility, *Journal of Fuzzy Sets and Systems* (1): 3–28.

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.

Apêndice A

Conceitos de Grafos Clássicos

Neste apêndice são apresentados os principais conceitos sobre grafos clássicos, isto é, grafos *crisp*. Tais conceitos são indispensáveis para uma melhor compreensão desta tese. Para um estudo mais completo é sugerido o Capítulo 2 de Ahuja et al. (1993) e os capítulos subsequentes com os problemas relativos a grafos. Outras referências utilizadas foram: Bazarra et al. (1990), Gondran e Minoux (1984) e Goldbarg e Luna (2000).

Definição 40 *Um grafo, denotado por $G = (V, A)$, é definido como um conjunto V cujos elementos são chamados nós (ou vértices) e um conjunto A cujos elementos $a \in A$ são pares ordenados de elementos de V (nós), chamados arcos (ou arestas).*

Definição 41 *Arco é 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)

Definição 42 *Aresta é um par não ordenado (i, j) . Neste caso, se a_{ij} é uma aresta, tem-se necessariamente que $a_{ij} = a_{ji}$ (Figura A.2).*

Definição 43 *Um grafo $G=(V,A)$ é dito orientado (direcionado) se seus arcos possuírem orientação (Figura A.3), caso contrário é dito não-orientado (Figura A.4).*

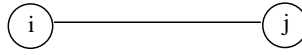
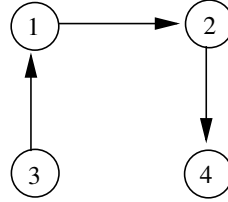
Figura A.2: Exemplo de uma aresta (i,j) 

Figura A.3: Exemplo de um grafo orientado

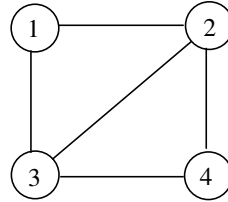


Figura A.4: Exemplo de um grafo não-orientado

Definição 44 *Dois arcos (arestas) são adjacentes se possuírem pelo menos um extremo em comum.*

Definição 45 *Um arco (aresta) cujos extremos coincidam é chamado laço (Figura A.5).*

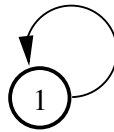


Figura A.5: Exemplo de um laço

Definição 46 *Caminho é uma seqüência orientada de nós e arcos,*

$$p = \{i, (i, j_1), j_1, (j_1, j_2), j_2, (j_2, j_3), j_3, \dots\},$$

sem repetições de nós (Figura A.6).

Através da Figura A.6 tem-se o seguinte caminho: $p = \{1, (1, 2), 2, (2, 3), 3, (3, 4)4\}$.

Definição 47 *O comprimento de um caminho, ou ciclo, é determinado pela quantidade de arcos existentes neste caminho ou ciclo.*

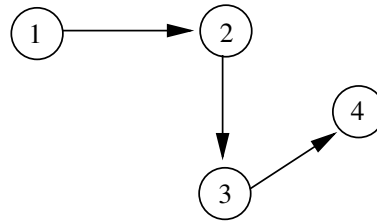


Figura A.6: Exemplo de um caminho

Por exemplo, o caminho p da figura anterior que sai do nó 1 e chega no 4 possui comprimento igual a 3.

Definição 48 *Capacidades de um arco (ou aresta) são valores referentes à quantidade mínima (l_{ij}) e máxima (u_{ij}) de fluxo (x_{ij}) que pode ser transportado pelo arco (ou aresta).*

Definição 49 *Fluxo é uma maneira de se enviar objetos sobre arestas (ou arcos) de um nó para outro do grafo. Associa-se números aos nós e arcos (ou arestas) de um grafo, normalmente designando uma capacidade (mínima e máxima) destes nós e arcos (ou arestas).*

Definição 50 *Custo de um arco 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.*

Definição 51 *Grau de um nó refere-se ao número de arcos (ou arestas) incidentes no referido nó.*

Por exemplo, na Figura A.4 os nós 2 e 3 têm grau 3.

Definição 52 *Um circuito é um caminho fechado (Figura A.7).*

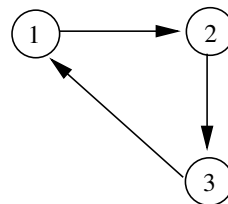


Figura A.7: Exemplo de um circuito

Definição 53 *Cadeia é uma seqüência não orientada de nós e arestas, sem repetições de nós.*

Definição 54 *Ciclo é uma cadeia fechada (Figura A.8).*

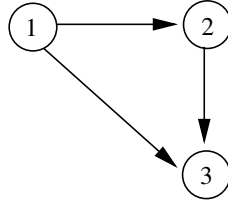


Figura A.8: Exemplo de um ciclo

Definição 55 *Um grafo $G = (V, A)$ é acíclico se não for possível formar ciclos com qualquer subsequência composta com os elementos de V e de A .*

Definição 56 *Em um problema de fluxo 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.9, o nó 1 é um nó origem.*

Definição 57 *Em um problema de fluxo 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.9, o nó 6 é um nó destino.*

Definição 58 *Em um problema de fluxo 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.9, os nós 2, 3, 4 e 5 são nós de transbordo.*

Definição 59 *Conjunto de corte $[X, \bar{X}]$: Seja um grafo $G = (V, A)$. Seja $X \subset V$ e $\bar{X} = V - X$. $[X, \bar{X}] = \{(i, j) \in A | i \in X, j \in \bar{X}\}$, ou seja, um conjunto de corte é formado pelos arcos que conectam X a $\bar{X}$ (Figura A.10).*

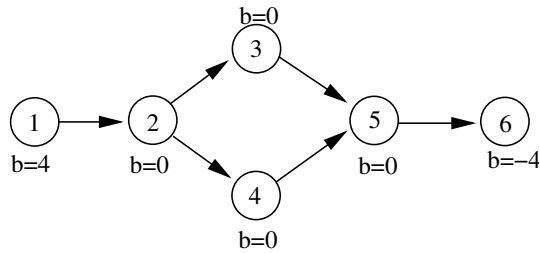
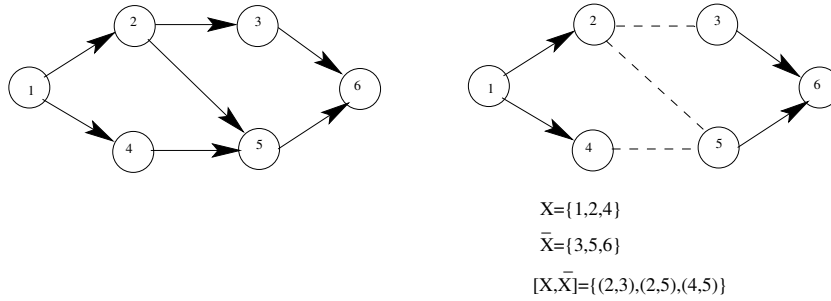


Figura A.9: Exemplo de nós capacitados

Figura A.10: Exemplo de um conjunto de corte $[X, \bar{X}]$

Definição 60 Dado um grafo $G = (V, A)$, um subgrafo é um grafo $G' = (V', A')$ tal que $V' \subseteq V$ e $A' \subseteq A$.

Definição 61 Dado um grafo $G = (V, A)$, um subgrafo gerador é um subgrafo $G' = (V', A')$ tal que $V' \equiv V$.

Definição 62 Um grafo $G = (V, A)$ é conexo se para todo par de vértices i e j que pertencem a V existe pelo menos um caminho unindo-os. Caso contrário é dito desconexo (Figura A.11).

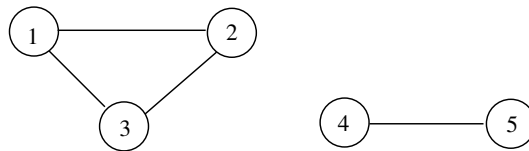


Figura A.11: Exemplo de um grafo desconexo

Definição 63 Uma árvore é um subgrafo conexo e sem ciclos.

Definição 64 Seja um grafo $G = (V, A)$. Uma árvore geradora é um subgrafo gerador de G que não possui ciclos.

Definição 65 *Um grafo completo é um grafo simples em que todo vértice é adjacente a todos os outros vértices. O grafo completo de n vértices é freqüentemente denotado por K_n .*

Apêndice B

Problemas de Grafos Clássicos

De modo geral, os problemas em grafos são aqueles em que a decisão está associada a sua estrutura (arcos e nós). A seguir são descritos o problemas clássicos que foram adaptados, nesta tese, para o problema fuzzy.

B.1 Problema de caminho mínimo

Este problema consiste em encontrar um caminho de custo mínimo entre dois nós específicos do grafo. Existem diversos algoritmos para este problema, sendo os algoritmos de Dijkstra e Ford-Moore-Bellman os mais conhecidos. Porém, como nesta tese foi usado o algoritmo de Ford-Moore-Bellman (Bellman 1958), este será descrito.

Algoritmo de Ford-Moore-Bellman

Informações:

V : conjunto dos nós;

it : contador de iterações;

r : número de nós;

l_{ji} : custo do arco (j, i) ;

c_i^{it} : custo do caminho entre os nós 1 e i na iteração it ; e

Γ_i^{-1} : conjunto dos nós predecessores de i .

Algoritmo

PASSO 1: Inicialização das variáveis

1. $c_1^0 = 0$
2. $c_i^0 = \infty, i = 2, 3, \dots, r$
3. $it \leftarrow 1$.

PASSO 2: Determinação dos caminhos das próximas iterações

1. $c_1^{it} = 0$;
2. $\forall i \in V, i = 2, 3, \dots, r$, faça:
 - $c_i^{it} = \min(c_i^{it-1}, \min_{j \in \Gamma_i^{-1}}(c_j^{it-1} + l_{ji}))$;

PASSO 3: Critério de parada

- Se $c_i^{it} = c_i^{it-1}, \forall i \in V \Rightarrow \text{FIM}$
- Senão
 - Se $it \leq r - 1 \Rightarrow \text{Passo 2}$
 - Senão existe um circuito com custo negativo $\Rightarrow \text{FIM}$

Este algoritmo possui a vantagem de ser aplicado em redes com arcos de custos negativos, contrário ao de Dijkstra.

B.2 Problemas de fluxo máximo

Considere 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, pode-se definir uma função $\phi(i, j)$ que corresponde ao fluxo do arco. Esta função tem que satisfazer 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; e
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.

Logo este problema tem por objetivo determinar o maior fluxo possível que pode ser enviado nos arcos do grafo, considerando as limitações de capacidade dos mesmos (obter o maior valor para $\phi(i, j)$).

Outras definições e a teoria envolvida relativa ao assunto podem ser vistas em Bazarra et al. (1990) e Ahuja et al. (1993).

B.3 Problemas de fluxo de custo mínimo (PFCM)

Este problema corresponde à necessidade de determinar o atendimento da demanda em uma rede pagando o mínimo possível pelo tráfego. Normalmente o fluxo a transitar é conhecido e existem restrições para o máximo e o mínimo fluxo admissível nos arcos, bem como um valor de custo associado ao trânsito em cada arco por unidade de fluxo (c_{ij}). Os dois algoritmos utilizados são o Simplex para redes e o *Out-of-Kilter*. Os problemas de caminho mínimo e fluxo máximo são casos especiais do PFCM (veja: Goldberg e Luna (2000)).

Como um dos algoritmos propostos nesta tese foi uma adaptação do método do Big-M, a seguir, o referido algoritmo é apresentado.

B.3.1 Algoritmo do Big-M

Este algoritmo tem por finalidade encontrar uma solução factível inicial para o algoritmo Fase II (descrito na seção seguinte).

Informações sobre o algoritmo proposto:

V : conjunto dos nós;

b_k : demanda do nó k ;

w_i : potencial do nó i ;

l_{ij} : limitante inferior do arco (i, j) (neste caso é zero);

L_{ij} : limitante superior do arco (i, j) ;

x_{ij} : fluxo que percorre o arco (i, j) ;

c_{ij} : custo de cada unidade de fluxo que percorre o arco (i, j) .

Algoritmo

Seja um grafo orientado $G = (V, A)$.

PASSO 1: Construção do grafo artificial (grafo original acrescido do nó artificial e de seus respectivos arcos, também artificiais)

- Criar um nó artificial $(n + 1)$
- Criar n arcos artificiais unindo os n nós do grafo original ao nó artificial $(n + 1)$, tal que:

- se $b_k > 0, \forall k \in V \rightarrow$ criar um arco artificial na direção de k para $(n+1)$;
- se $b_k < 0, \forall k \in V \rightarrow$ criar um arco artificial na direção de $(n+1)$ para k ; e
- se $b_k = 0, \forall k \in V \rightarrow$ criar um arco artificial na direção de k para $(n+1)$, porém poderia ser na direção contrária.

PASSO 2: Definir o nó $(n + 1)$ como raiz e associar o custo ∞ a todos os arcos artificiais.

Para a implementação, o valor ∞ pode ser substituído pela soma dos custos de todos os arcos da rede acrescido de uma unidade.

PASSO 3: Aplicar o método simplex especializado

1. Atribuição do fluxo inicial:

- para os arcos do problema original, fixar o fluxo inicial no limitante inferior (no problema em questão fixar em zero);
- para os arcos artificiais designar fluxo de tal forma que o problema seja factível, ou seja, atribuir o fluxo b_k no arco artificial que tem k como um dos seus extremos; e
- a base inicial é formada pelos arcos artificiais.

2. Cálculo dos potenciais:

Para todos os nós do grafo calcular os seus potenciais da seguinte forma:

- $w_{n+1} = 0$ (ao nó artificial $(n + 1)$ é atribuído valor zero).
- Utilizando os arcos da base, calcular os demais potenciais da seguinte forma:

$$w_i - w_j = c_{ij}, \forall i, j \in V^* \text{ (} V^*: \text{ conjunto dos nós do grafo artificial e } c_{ij}: \text{ custo do arco (i,j))}.$$

3. Cálculo dos custos relativos:

Para todos os arcos não-básicos, calcular os custos relativos da seguinte forma:

- $\hat{c}_{ij} = z_{ij} - c_{ij} = w_i - w_j - c_{ij}, \forall (i, j) \in A^* \text{ (} A^*: \text{ conjunto dos arcos não básicos)}.$

4. Verificação do arco candidato a entrar na base:

- Entrará na base o arco não-básico que possuir o maior custo relativo positivo, caso haja empate, escolher um deles.

5. Verificação do arco candidato a deixar a base:

- Encontrar um ciclo formado entre o arco candidato a entrar na base e outros arcos pertencentes à base, sendo que deixará a base o arco que bloquear o tráfego de fluxo naquele ciclo. O bloqueio é calculado da seguinte forma:

- Para todo arco que possui orientação contrária à do arco candidato a entrar na base, calcular:

$$\delta_{ij}^- = x_{ij} - l_{ij}$$

- Para todo arco que possuir a mesma orientação do arco candidato a entrar na base, calcular:

$$\delta_{ij}^+ = L_{ij} - x_{ij}$$

- Calcular $\Delta = \min\{\delta_{ij}^-, \delta_{ij}^+\}$

6. Redefinir o fluxo ao longo do ciclo encontrado no passo anterior da seguinte forma:

- Se (i, j) tiver a mesma direção do arco candidato a entrar na base atribuir:

$$x_{ij} = x_{ij} + \Delta$$

- Se (i, j) tiver direção contrária a do arco candidato a entrar na base atribuir:

$$x_{ij} = x_{ij} - \Delta$$

7. Critério de parada:

- Enquanto existir mais de um arco artificial na base, repetir do Passo 2 ao Passo 6 deste passo.

PASSO 4: Construção da solução factível:

- Se o único arco artificial que estiver na base possuir fluxo não-nulo $\Rightarrow$ problema sem solução.

- Caso contrário, elimine o nó-artificial e todos os arcos incidentes nele e construa a solução factível para aplicar o algoritmo Fase II.

B.3.2 Algoritmo Fase II

Este algoritmo tem como objetivo encontrar a solução ótima do problema, partindo da solução factível encontrada pelo Big-M. Este é similar ao Passo 3 do algoritmo anterior, chamado *simplex especializado*.

Algoritmo

PASSO 1: Definir um nó raiz, w_r , atribuir $w_r = 0$ e iniciar o algoritmo com a solução inicial factível obtida através da aplicação do Big-M

PASSO 2: Calcular os potenciais e os custos relativos (similar aos itens 2 e 3 do Passo 3 do Big-M).

PASSO 3: Enquanto existir pelo menos um arco (i, j) tal que $(z_{ij} - c_{ij}) > 0$ faça:

- Determinar o arco candidato a entrar na base (semelhante ao item 4 do Passo 3 do Big-M).
- Determinar o arco candidato a deixar a base (semelhante ao item 5 do Passo 3 do Big-M).
- Recalcular o fluxo na rede (semelhante ao item 6 do Passo 3 do Big-M).
- Recalcular os potenciais e os custos (semelhante aos itens 2 e 3 do Passo 3 do Big-M).

PASSO 4: Valor da função objetivo: $f = \sum_{(i,j) \in A} c_{ij}x_{ij}$

B.4 Problema de coloração de grafos

Uma coloração dos vértices de um grafo não-orientado é a atribuição de cores aos vértices que tenham a seguinte propriedade: as pontas de cada aresta têm cores diferentes.

Logo, o problema da coloração de vértices consiste em: dado um grafo não-orientado, encontrar uma coloração dos vértices com número mínimo de cores (maiores detalhes: Chartrand e Oellermann (1993)).