

FLÁVIO FERNANDES BARBOSA SILVA

Desvendando a Lógica Fuzzy



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE MATEMÁTICA

2011

FLÁVIO FERNANDES BARBOSA SILVA

Desvendando a Lógica Fuzzy

Dissertação apresentada ao Programa de Pós-Graduação em Matemática da Universidade Federal de Uberlândia, como parte dos requisitos para obtenção do título de **MESTRE EM MATEMÁTICA**.

Área de Concentração: Matemática.

Linha de Pesquisa: Análise Numérica.

Orientador: Prof. Dr. César Guilherme de Almeida.

UBERLÂNDIA - MG

2011

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU , MG, Brasil

S586d Silva, Flávio Fernandes Barbosa, 1984-
Desvendando a lógica Fuzzy [manuscrito] / Flávio Fernandes Barbosa
Silva. - 2011.
129 f. : il.

Orientador: César Guilherme de Almeida.

Dissertação (mestrado) – Universidade Federal de Uberlândia, Programa de Pós-Graduação em Matemática.
Inclui bibliografia.

1. Análise numérica - Teses. 2. Conjuntos difusos - Teses. I. Almeida, César Guilherme de. II. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Matemática. III. Título.

CDU: 519.6



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE MATEMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA

Av. João Naves de Ávila, 2121, Bloco 1F, Sala 1F 152
Campus Santa Mônica, Uberlândia - MG, CEP 38400-902

ALUNO: Flávio Fernandes Barbosa Silva.

NÚMERO DE MATRÍCULA: 100092.

ÁREA DE CONCENTRAÇÃO: Matemática.

LINHA DE PESQUISA: Análise Numérica.

PÓS-GRADUAÇÃO EM MATEMÁTICA: Nível Mestrado.

TÍTULO DA DISSERTAÇÃO: Desvendando a Lógica Fuzzy.

ORIENTADOR: Prof. Dr. César Guilherme de Almeida.

Esta dissertação foi **APROVADA** em reunião pública realizada na Sala Multiuso da Faculdade de Matemática, Bloco 1F, Campus Santa Mônica, em 06 de maio de 2011, às 8h00min, pela seguinte Banca Examinadora:

NOME

Prof. Dr. César Guilherme de Almeida

UFU - Universidade Federal de Uberlândia

Prof. Dr. Laécio Carvalho de Barros

Unicamp - Universidade Estadual de Campinas

Profa. Dra. Márcia Aparecida Fernandes

UFU - Universidade Federal de Uberlândia

ASSINATURA

Uberlândia-MG, 06 de maio de 2011.

Dedicatória

A Deus por me mostrar o caminho e por me dar a oportunidade de, tentar e conseguir, alcançar os meus objetivos.

Ao meu amigo Uberte Teixeira por ter me acolhido em teu lar e em teu coração com um grande carinho e confiança e por ser umas das pessoas mais verdadeiras e raras que conheci.
À minha mãe por sempre estar ao meu lado.

Agradecimentos

Ao Prof. Dr. César Guilherme de Almeida pela amizade e orientação, pela confiança e por ter acreditado em mim na realização deste trabalho.

A Profa. Dra. Rosana Sueli da Motta Jafelice pela co-orientação em nosso trabalho.

Aos professores Laécio Carvalho de Barros e Márcia Aparecida Fernandes, pelas correções e sugestões.

A minha família pelo carinho.

A minha namorada e amiga Marla Francení pelo seu amor, força e confiança depositados em mim.

Aos meus amigos do curso de mestrado.

A todos os meus amigos motoristas que compartilharam comigo as suas viagens me ajudando sempre na estrada.

Aos professores do Programa de Pós Graduação em Matemática da UFU.

Ao professor Carlos Alberto Raposo da UFSJ.

A minha segunda família de São João Del Rei que me acolheu com muito carinho, em especial a tia Lurdes que me tratou como filho e que hoje está ao lado de Deus.

A Deus por ter colocado todas essas pessoas em meu caminho.

SILVA, F. F. B. *Desvendando a Lógica Fuzzy*. 2011. (117 pág) p. Dissertação de Mestrado, Universidade Federal de Uberlândia, Uberlândia-MG.

Resumo

O principal objetivo desta dissertação é o de construir um algoritmo que execute os procedimentos referentes ao Sistema Baseado em Regras *Fuzzy* (SBRF). Para este fim, primeiramente, será estudada a teoria dos conjuntos *fuzzy* juntamente com as definições de cada componente do SBRF. Depois, serão exibidos os códigos computacionais que permitirão compreender, passo-a-passo, a técnica de resolução de vários problemas que utilizam esta poderosa ferramenta. Tal ferramenta é bastante difundida no *software* MATLAB; o nosso intuito é torná-la mais acessível, através da apresentação de códigos computacionais que foram desenvolvidos em uma linguagem computacional que não está vinculada a um específico pacote computacional. A programação das rotinas que compõem o algoritmo podem ser feitas, por exemplo, no Octave, que é um *software* livre similar ao MATLAB. Uma vantagem do algoritmo proposto, em relação aos códigos elaborados no MATLAB, diz respeito à facilidade de se fazer ajustes dinâmicos nos parâmetros do SBRF e de se integrar as rotinas que foram desenvolvidas a qualquer código computacional, escrito em uma dada linguagem de programação, especificamente desenvolvido para simular a solução de um problema modelado com a teoria dos conjuntos fuzzy, sem a necessidade de se fazer adaptações que permitiriam o acesso ao ferramental *fuzzy* (*Fuzzy Logic toolbox*) do MATLAB.

Palavras-chave: (conjuntos *Fuzzy*, Inferência de Mamdani, Inferência de Takagi-Sugeno).

SILVA, F. F. B. *Title - italics and english*. 2011. (117 pages) p. M. Sc. Dissertation, Federal University of Uberlândia, Uberlândia-MG.

Abstract

The purpose of this Master Dissertation is to build an algorithm which executes the Fuzzy Rule-Based System (FRBS) procedures. To reach this aim, it will be firstly studied the basic concepts of fuzzy set theory along with all FRBS components. After that, the computer codes that exhibit step-by-step some techniques for solving several problems using this powerful tool are going to show. Such a tool is widespread in Matlab. Promoting and spreading the accessibility to that tool is the intention of this work. It will be made through the presentation of several codes, which were developed in this dissertation. These codes do not depend on a specific computer package. The algorithm can be implemented in any existing programming language like C, for example. The programming of the algorithm can be made in Octave environment, too. Octave is similar to MATLAB; besides, it is a free software. Making dynamic adjustments of the FRBS parameters is an advantage of the algorithm developed here compared with MATLAB one. Note that any adjustments can be easily made while the code is running.

Keywords: (Fuzzy sets, Mamdani inference, Takagi-Sugeno inference).

Lista de Figuras

1.1	Representação da união dos conjuntos <i>fuzzy</i> A e B . [18]	6
1.2	Representação da intersecção dos conjuntos <i>fuzzy</i> A e B . [18]	6
1.3	Representação do complemento de um conjunto <i>fuzzy</i> A . [18]	7
1.4	Representação da União padrão. [18]	8
1.5	Representação da Soma algébrica. [18]	9
1.6	Representação da Soma limitada. [18]	10
1.7	Representação da União drástica. [18]	10
1.8	Representação da Intersecção padrão. [18]	11
1.9	Representação do produto algébrico. [18]	12
1.10	Representação da Diferença limitada. [18]	13
1.11	Representação da Intersecção drástica. [18]	13
3.1	Variáveis Linguísticas. [18]	22
3.2	Esquema para um sistema de controle humano na tarefa de lavar roupa.	23
3.3	Sistemas Baseados em Regras <i>fuzzy</i> . [7]	24
3.4	Saídas parciais do controlador fuzzy de Mamdani	26
3.5	Saída final do controlador fuzzy de Mamdani	26
3.6	Defuzzificador centro de gravidade	27
3.7	Saída do controlador <i>fuzzy</i> TSK para o Exemplo 3.2	29
4.1	Algoritmo Sistema Baseado em Regras Fuzzy (ASBRF)	31
4.2	Função de pertinência trapezoidal.	32
4.3	Função de pertinência triangular.	33
4.4	Função de pertinência gaussiana para valores α distintos.	34
4.5	Saída final do programa ASBRF para o método de Mamdani.	41
4.6	Defuzzificador centro de gravidade	42
4.7	Função de pertinência da Quantidade de Água (A)	48

4.8	Função de pertinência do Tempo de exposição ao Sol(S)	48
4.9	Função de pertinência da Vitalidade das Violetas(V).	48
4.10	Saída final do controlador fuzzy de Mamdani	50
5.1	(a): Antecedentes. (b) Consequentes y_α e y_β	53
5.2	Dados, curvas dos modelos determinístico e <i>fuzzy</i>	54
5.3	Funções de pertinência para a variável de entrada resistência do solo à penetração para o solo não preparado do tipo III.	58
5.4	Funções de pertinência para a variável de entrada teor de água (umidade) para o solo não preparado do tipo III.	58
5.5	Resistência à penetração do Solo	63
5.6	Teor de água no Solo(Umidade)	63
5.7	Funções de pertinência da Carga viral (V).	70
5.8	Função de pertinência do nível de $CD4^+$	70
5.9	Valores da taxa de transferência defuzzificados.	71
5.10	λ como função do $CD4^+$ ($v = 0.1$).	72
5.11	λ como função da carga viral ($c = 1$).	72

(Listar as figuras e suas legendas - Opcional)

Sumário

Resumo	vii
Abstract	viii
Lista de Figuras	x
Introdução	1
1 Conjuntos Fuzzy	3
1.1 Conjuntos Fuzzy	3
1.2 Operações entre conjuntos <i>fuzzy</i>	5
1.3 Normas Triangulares	7
2 Relações Fuzzy	14
2.1 Relações Fuzzy	14
2.1.1 Composição de Relações Fuzzy	17
3 Sistema Baseado em Regras Fuzzy	21
3.1 Regras e inferência <i>fuzzy</i>	21
3.2 Variáveis Linguísticas	21
3.2.1 Termos Linguísticos	22
3.3 Sistemas Baseados em Regras Fuzzy	22
3.3.1 Processador de Entrada (Fuzzificação)	23
3.3.2 Base de Regras	24
3.3.3 Máquina de Inferência <i>Fuzzy</i>	24
3.3.4 Defuzzificação	25
3.4 Método de Inferência de Mamdani	25
3.5 Métodos de Defuzzificação	26
3.5.1 Centro de Gravidade ($G(C)$), Centróide ou Centro de Área	26

3.6	Método de Inferência de Takagi - Sugeno - Kang (TSK)	27
4	Algoritmo do Sistema Baseado em Regras Fuzzy (ASBRF)	30
4.1	Estrutura do ASBRF	31
4.1.1	Entrada de dados	31
4.1.2	Parâmetros	32
4.1.3	Base de Regras	36
4.1.4	DADOS	39
4.1.5	Processamento de dados	40
4.1.6	Defuzzificação	42
4.1.7	Algoritmo (ASBRF - Mamdani) - Estrutura Computacional	44
4.1.8	Algoritmo (ASBRF - Sugeno) - Estrutura Computacional	45
4.1.9	Exemplo - Vitalidade das Violetas	46
5	Aplicações	51
5.1	Modelo 1 - Decaimento de Fármaco	51
5.2	Modelo 2: Modelo <i>Fuzzy</i> para a Densidade do Solo	55
5.3	Modelo 3: Conversão de Assintomático para Sintomático. Modelos <i>Fuzzy</i> com λ Dependendo do Nível de CD4+ e da Carga Viral	66
5.3.1	Informações Médicas sobre HIV	66
5.3.2	O Modelo Clássico	68
5.3.3	O Modelo <i>Fuzzy</i>	69
6	Conclusão	73
	Referências Bibliográficas	74
	Apêndice	77

Introdução

A Teoria dos Conjuntos *Fuzzy*, recente do ponto de vista de historiografia, vem se desenvolvendo e ganhando espaço e, cada vez mais, está sendo usada como ferramenta para formulação de modelos nos vários campos das ciências. Essa teoria foi introduzida, por volta do ano de 1965, pelo matemático Lotfi A. Zadeh, e o seu desenvolvimento e suas aplicações vêm apresentando uma evolução muito rápida. Podemos dizer que esta teoria já tem um lugar de destaque, com suas aplicações práticas cada vez mais bem sucedidas.

Nesta dissertação apresentaremos o Algoritmo do Sistema Baseado em Regras Fuzzy (AS-BRF), que foi desenvolvido com o objetivo de executar os procedimentos referentes ao Sistema Baseado em Regras Fuzzy (SBRF). Estes procedimentos são bem conhecidos dos usuários do “ferramental *fuzzy*” do *software* MATLAB, denominado *Fuzzy Logic Toolbox*, que é empregado em diversas modelagens matemáticas que utilizam a teoria dos conjuntos *fuzzy*.

As rotinas que apresentaremos podem ser desenvolvidas em qualquer linguagem de programação, tais como as linguagens **C** e Fortran.

Uma vantagem do algoritmo que está sendo proposto neste trabalho, em relação aos códigos elaborados no MATLAB, diz respeito à facilidade de se fazer ajustes dinâmicos no SBRF, tais como: i) alteração de parâmetros relacionados à construção das funções de pertinência; isto permitiria a alteração do domínio da função de pertinência e, até mesmo, a alteração do formato desta função; ii) alteração da quantidade de regras do sistema *fuzzy*. Estes ajustes seriam feitos durante a execução de um código computacional, sem a necessidade de interrupções para a inclusão de novas informações.

Outra vantagem que o algoritmo desenvolvido neste trabalho nos oferece é a utilização de funções mais gerais para a saída da base de regras do método de Takagi-Sugeno. No *software* MATLAB, em relação a este método, somente são usadas, como consequentes para a base de regras, funções constantes ou lineares.

A dissertação está estruturada da seguinte maneira:

- Os Capítulos 1 e 2 apresentam as definições básicas da teoria dos conjuntos *fuzzy* que

utilizamos no decorrer deste trabalho. As principais referências usadas na construção desses capítulos foram [3], [8] e [18].

- O Capítulo 3 introduz o conceito de variáveis linguísticas e o Sistema Baseado em Regras *Fuzzy*. Concentraremos nosso estudo no funcionamento de cada componente deste sistema, utilizando os métodos de inferência de Mamdani e Takagi-Sugeno-kang, para mais adiante elaborarmos o Algoritmo (ASBRF). Usamos a referência [3] no desenvolvimento desse capítulo.
- Após os estudos realizados no Capítulo 3, é possível apresentar a construção do Algoritmo (ASBRF). Isto é feito no Capítulo 4, no qual apresentamos um fluxograma que representa o funcionamento do ASBRF. Além disto, ao longo deste capítulo, defini-se cada componente do algoritmo, fazendo-se referência às rotinas que compõe tal componente. Por fim, apresentamos, para a utilização dos métodos de inferência de Mamdani e Sugeno, a estrutura computacional do ASBRF.
- O Capítulo 5 apresenta algumas aplicações que foram realizadas em outros trabalhos [3, 4, 7]. Essas aplicações validarão o ASBRF desenvolvido nesta dissertação.

Flávio Fernandes Barbosa Silva
Uberlândia-MG, 06 de maio de 2011.

Capítulo 1

Conjuntos Fuzzy

Neste capítulo apresentaremos os conceitos e ferramentas básicas, com alguns exemplos da Lógica Fuzzy estudados em [3], como instrumento de aplicações que utilizaremos no decorrer deste trabalho.

1.1 Conjuntos Fuzzy

Definição 1.1 *Seja U um conjunto e A um subconjunto de U . A função característica de A é dada por*

$$\chi_A(x) = \begin{cases} 1 & \text{se } x \in A \\ 0 & \text{se } x \notin A \end{cases}$$

Desta forma, χ_A é uma função cujo domínio é U e a imagem está contida no conjunto $\{0,1\}$, com $\chi_A(x) = 1$ indicando que o elemento x está em A , enquanto $\chi_A(x) = 0$ indica que x não é elemento de A . Assim, a função característica descreve completamente o conjunto A já que tal função indica quais elementos do conjunto universo U são elementos também de A . Entretanto, do ponto de vista das aplicações, existem casos em que a pertinência entre elementos e conjuntos não é precisa, isto é, não sabemos dizer se um elemento pertence efetivamente a um conjunto ou não. Neste caso é plausível dizer qual elemento do conjunto se enquadra “melhor” ao termo que caracteriza o subconjunto. Por exemplo, consideremos o subconjunto dos números reais próximos de 2.

$$A = \{x \in \mathbb{R} : x \text{ é próximo de } 2\}.$$

Pergunta: O número 7 e o número 2,001 pertencem a A ?

A resposta a esta pergunta é incerta pois não sabemos até que ponto podemos dizer objetivamente quando um número está próximo de 2. A única afirmação razoável, neste caso, é que 2,001 está mais próximo de 2 do que 7.

A seguir vamos iniciar as formalizações matemáticas dos conceitos de Lógica *Fuzzy* que serão tratados neste texto, começando com o de subconjunto *fuzzy*.

Definição 1.2 *Seja U um conjunto (clássico); um subconjunto fuzzy F de U é caracterizado por uma função*

$$\varphi_F : U \rightarrow [0, 1],$$

pré-fixada, chamada função de pertinência do subconjunto fuzzy F . O índice F na função de pertinência é usado em analogia à função característica de subconjunto clássico, conforme a Definição 1.1. O valor $\varphi_F \in [0, 1]$ indica o grau de pertinência com que o elemento x de U está no conjunto F ; $\varphi_F(x) = 0$ e $\varphi_F(x) = 1$ indicam, respectivamente, a não pertinência e a pertinência completa de x ao conjunto fuzzy F .

Do ponto de vista formal, a definição de subconjunto fuzzy foi obtida simplesmente ampliando-se o contra-domínio da função característica que é o conjunto $\{0, 1\}$, para o intervalo $[0, 1]$. Nesse sentido, podemos dizer que um conjunto clássico é um caso particular de um dado conjunto fuzzy, cuja função de pertinência φ_F é uma função característica χ_F .

Exemplo 1.1 (Números próximos de 2). *Considere o subconjunto F dos números reais próximo de 2:*

$$F = \{x \in \mathbb{R} : x \text{ é próximo de } 2\}$$

Se definirmos a função $\varphi_F : \mathbb{R} \rightarrow [0, 1]$, que associa a cada x real o valor de proximidade ao ponto 2 pela expressão

$$\varphi_F(x) = \begin{cases} (1 - |x - 2|) & \text{se } 1 < x < 3 \\ 0 & \text{se } x \notin [1, 3] \end{cases},$$

então o subconjunto F dos pontos próximos de 2, caracterizado por φ_F , é tal que $\varphi_F(2,001) = 0,999$ e $\varphi_F(7) = 0$. Neste caso dizemos que $x = 2,001$ é um ponto próximo de 2 com grau de proximidade 0,999 e $x = 7$ não é próximo de 2.

Por outro lado, alguém poderia sugerir outra função de proximidade a 2. Por exemplo, se a função de proximidade a 2 foi definida por

$$\nu_F(x) = \exp[-(x - 2)^2],$$

com $x \in \mathbb{R}$, então os elementos do conjunto F , caracterizado pela função ν_F , teriam outros graus de pertinência: $\nu_F(2,001) = 0,999999$ e $\nu_F(7) = 1,388 \times 10^{-11}$.

Como podemos ver, a caracterização de proximidade é subjetiva e depende da função de pertinência que pode ser dada de uma infinidade de maneiras diferentes, dependendo de como se quer avaliar o termo “próximo”. Observe que poderíamos também definir próximo de 2 por um conjunto clássico por uma função de $\varphi_{\epsilon,F}$, considerando, por exemplo, um valor de ϵ suficientemente pequeno e a função característica do intervalo aberto $(2 - \epsilon, 2 + \epsilon)$, conforme a expressão abaixo

$$\varphi_{\epsilon,F}(x) = \begin{cases} 1 & \text{se } |x - 2| < \epsilon \\ 0 & \text{se } |x - 2| > \epsilon \end{cases}.$$

Note que próximo de 2 significa estar numa vizinhança pré determinada de 2. A subjetividade está exatamente na escolha do raio da vizinhança. Especificamente, neste caso todos os valores desta vizinhança estão próximos de 2 com o mesmo grau de pertinência que é 1.

1.2 Operações entre conjuntos fuzzy

Nesta seção estudaremos as operações típicas de conjuntos como união, intersecção e complementação.

Sejam A e B dois subconjuntos fuzzy de U , com funções de pertinência indicadas por φ_A e φ_B , respectivamente. Dizemos que A é subconjunto fuzzy de B , e escrevemos $A \subset B$, se $\varphi_A \leq \varphi_B$ para todo $x \in U$.

Lembramos que a função de pertinência do conjunto vazio ($\emptyset$) é dada por $\varphi_{\emptyset}(x) = 0$, enquanto que o conjunto universo U tem função de pertinência $\varphi_U(x) = 1$, para todo $x \in U$. Sejam A e B subconjuntos clássicos de U representados pelas funções características φ_A e φ_B , respectivamente. Os conjuntos

$$A \cup B = \{x \in U; \quad x \in A \quad \text{ou} \quad x \in B\},$$

$$A \cap B = \{x \in U; \quad x \in A \quad \text{e} \quad x \in B\},$$

$$A' = \{x \in U; \quad x \notin A\}.$$

Definição 1.3 (União) Sejam A e B conjuntos fuzzy. A união entre A e B é o subconjunto fuzzy de U cuja função de pertinência é dada por

$$\varphi_{(A \cup B)}(x) = \max\{\varphi_A(x), \varphi_B(x)\}, \quad x \in U.$$

A função de pertinência que representa o conjunto *fuzzy* união está representada na Figura 1.1

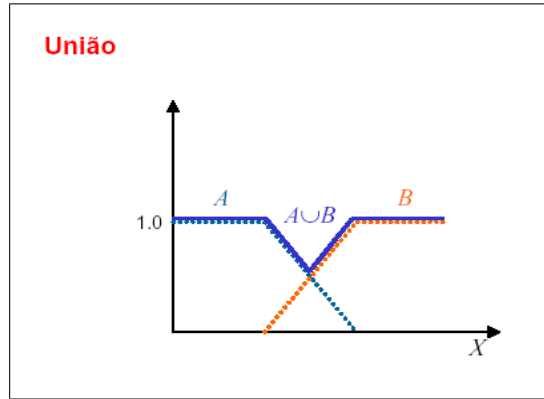


Figura 1.1: Representação da união dos conjuntos *fuzzy* A e B.[18]

Definição 1.4 (Intersecção). A intersecção entre A e B é o subconjunto *fuzzy* de U cuja função de pertinência é dada por

$$\varphi_{(A \cap B)}(x) = \min \{ \varphi_A(x), \varphi_B(x) \},$$

e está representada pela Figura 1.2

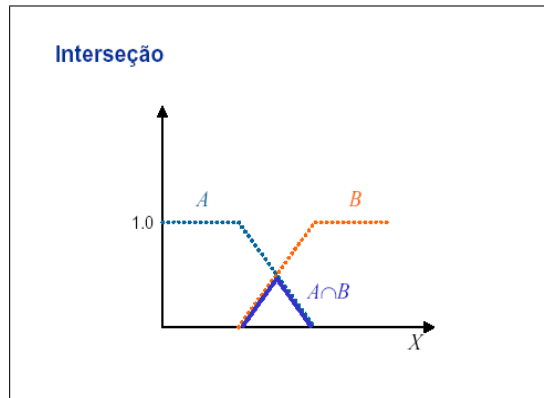


Figura 1.2: Representação da intersecção dos conjuntos *fuzzy* A e B.[18]

Definição 1.5 (Complementar de subconjuntos *fuzzy*). O complementar de A é o subconjunto *fuzzy* A' de U cuja função de pertinência é dada por

$$\varphi_{A'}(x) = 1 - \varphi_A(x),$$

e está representada pela Figura 1.3

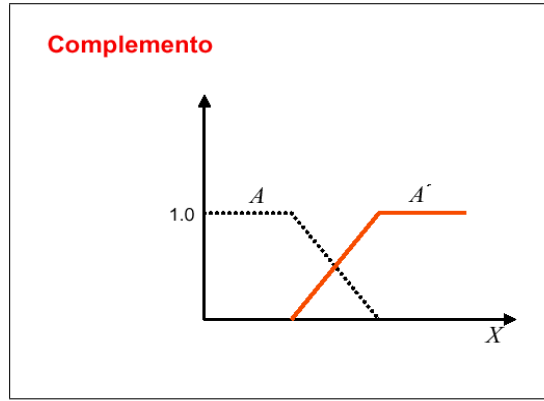


Figura 1.3: Representação do complemento de um conjunto *fuzzy* A . [18]

Exemplo 1.2 *Seja U um conjunto universo composto por pacientes de uma clínica, identificados pelos números 1, 2, 3, 4 e 5. Sejam A e B os conjuntos fuzzy que representam os pacientes com febre e mialgia, respectivamente. A Tabela 1.1, dada a seguir contém a união $A \cup B$, a intersecção $A \cap B$, o complemento A' e a intersecção $A \cap A'$. Observe que, diferentemente dos conjuntos clássicos, $A \cap A' \neq \emptyset$, de acordo com a Definição 1.5.*

Paciente	Febre: A	Mialgia: B	$A \cup B$	$A \cap B$	A'	$A \cap A'$
1	0.7	0.6	0.7	0.6	0.3	0.3
2	1.0	1.0	1.0	1.0	0.0	0.0
3	0.4	0.2	0.4	0.2	0.6	0.4
4	0.5	0.5	0.5	0.5	0.5	0.5
5	1.0	0.2	1.0	0.2	0.0	0.0

Tabela 1.1: União, intersecção e complementar dos conjuntos A e B

1.3 Normas Triangulares

As normas triangulares generalizam os operadores de união e intersecção e podem ser definidas da seguinte maneira:

Definição 1.6 *Uma conorma triangular (t -conorma) é uma operação binária $\nabla : [0, 1] \times [0, 1] \rightarrow [0, 1]$ satisfazendo:*

- *Comutatividade:* $x \nabla y = y \nabla x$
- *Associatividade:* $x \nabla (y \nabla z) = (x \nabla y) \nabla z$

- *Monotonicidade:* Se $x \leq y$ e $w \leq z$ então $x \nabla w \leq y \nabla z$
- *Condições de Fronteira:* $x \nabla 0 = x$, $x \nabla 1 = 1$.

Apresentaremos a seguir quatro exemplos de t-conorma: União padrão (operador max), Soma algébrica, Soma limitada e União drástica, e para cada uma delas mostraremos que vale as condições da Definição 1.6.

1. União padrão Figura (1.4) $\nabla : [0, 1] \times [0, 1] \longrightarrow [0, 1]$ com $x \nabla y = \max(x, y)$.

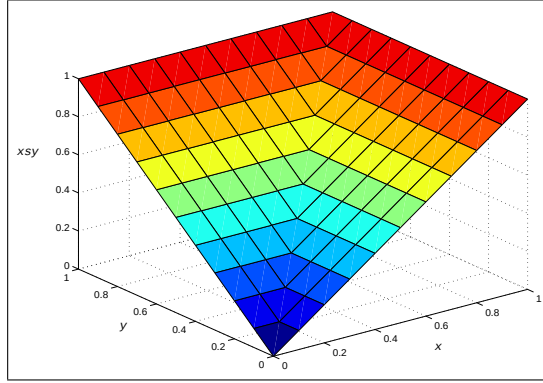


Figura 1.4: Representação da União padrão.[18]

- (a) Comutatividade: $x \nabla y = y \nabla x$.

De fato, pois $x \nabla y = \max(x, y) = \max(y, x) = y \nabla x$.

- (b) Associatividade: $x \nabla (y \nabla z) = (x \nabla y) \nabla z$.

Tem-se que, $x \nabla (y \nabla z) = x \nabla (\max(y, z)) = \max(x, \max(y, z))$ e $(x \nabla y) \nabla z = (\max(x, y)) \nabla z = \max(\max(x, y), z)$

Daí segue que $x \nabla (y \nabla z) = (x \nabla y) \nabla z$.

- (c) Monotonicidade: Se $x \leq y$ e $w \leq z$ então $x \nabla w \leq y \nabla z$.

Tem-se que, $x \nabla w = \max(x, w)$ e $y \nabla z = \max(y, z)$ e por hipótese $x \leq y$ e $w \leq z$, logo $\max(x, w) \leq \max(y, z)$, portanto $x \nabla w = \max(x, w) \leq \max(y, z) = y \nabla z$, ou seja, $x \nabla w \leq y \nabla z$.

- (d) Condições de Fronteira: $x \nabla 0 = x$, $x \nabla 1 = 1$.

Por hipótese tem-se que $0 \leq x \leq 1$, logo: $x \nabla 0 = \max(x, 0) = x$ e $x \nabla 1 = \max(x, 1) = 1$.

De (a),(b),(c) e (d) segue pela Definição 1.6 que a união padrão é uma t-conorma triangular.

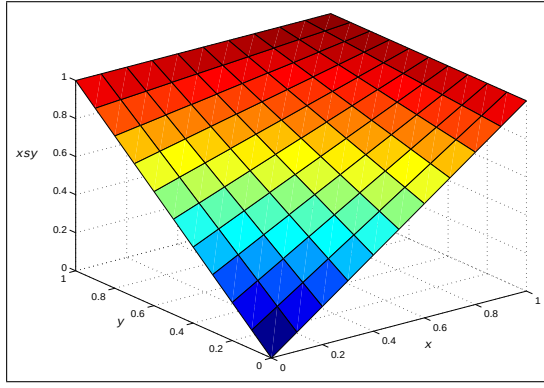


Figura 1.5: Representação da Soma algébrica.[18]

2. Soma Algébrica Figura(1.5) $\nabla : [0, 1] \times [0, 1] \longrightarrow [0, 1]$ com $x \nabla y = x + y - xy$.

(a) Comutatividade: $x \nabla y = y \nabla x$.

De fato, pois $x \nabla y = x + y - xy = y + x - yx = y \nabla x$.

(b) Associatividade: $x \nabla (y \nabla z) = (x \nabla y) \nabla z$.

Observe que, $x \nabla (y \nabla z) = x \nabla (y + z - yz) = x + y + z - yz - xy - xz + xyz$ e
 $(x \nabla y) \nabla z = (x + y - xy) \nabla z = x + y - xy + z - xz - yz + xyz$

Daí segue que $x \nabla (y \nabla z) = (x \nabla y) \nabla z$.

(c) Monotonicidade: Se $x \leq y$ e $w \leq z$ então $x \nabla w \leq y \nabla z$.

Tem-se que, $x \nabla w = x + w - xw$ e $y \nabla z = y + z - yz$ e por hipótese $x \leq y$ e $w \leq z$,
 logo $xw \leq yz$, portanto $x \nabla w = x + w - xw \leq y + z - yz = y \nabla z$, ou seja, $x \nabla w \leq y \nabla z$.

(d) Condições de Fronteira: $x \nabla 0 = x$, $x \nabla 1 = 1$.

Observe que, $x \nabla 0 = x + 0 - x.0 = x$ e $x \nabla 1 = x + 1 - x.1 = 1$.

De (a),(b),(c) e (d) segue pela Definição 1.6 que a soma algébrica é uma t-conorma triangular.

3. Soma Limitada Figura(1.6) $\nabla : [0, 1] \times [0, 1] \longrightarrow [0, 1]$ com $x \nabla y = \min(1; x + y)$.

(a) Comutatividade: $x \nabla y = y \nabla x$.

De fato, pois $x \nabla y = \min(1; x + y) = \min(1; y + x) = y \nabla x$.

(b) Associatividade: $x \nabla (y \nabla z) = (x \nabla y) \nabla z$.

Observe que, $x \nabla (y \nabla z) = x \nabla (\min(1; y + z)) = \min(1; x + y + z)$ e $(x \nabla y) \nabla z =$
 $(\min(1; x + y)) \nabla z = \min(1; x + y + z)$.

Daí segue que $x \nabla (y \nabla z) = (x \nabla y) \nabla z$.

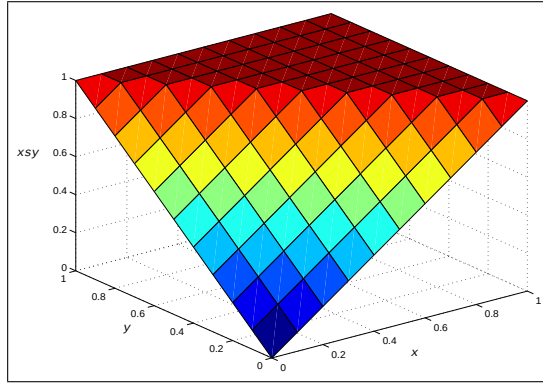


Figura 1.6: Representação da Soma limitada.[18]

(c) Monotonicidade: Se $x \leq y$ e $w \leq z$ então $x \nabla w \leq y \nabla z$.

Tem-se que, $x \nabla w = \min(1; x + w)$ e $y \nabla z = \min(1; y + z)$, por hipótese $x \leq y$ e $w \leq z$, segue que $x + w \leq y + z$, portanto $x \nabla w = \min(1; x + w) \leq \min(1; y + z) = y \nabla z$, ou seja, $x \nabla w \leq y \nabla z$.

(d) Condições de Fronteira: $x \nabla 0 = x$, $x \nabla 1 = 1$.

Por hipótese tem-se que $0 \leq x \leq 1$, logo: $x \nabla 0 = \min(1; x + 0) = \min(1; x) = x$ e $x \nabla 1 = \min(1; x + 1) = 1$.

De (a),(b),(c) e (d) segue pela Definição 1.6 que a soma limitada é uma t-conorma triangular.

4. União Drástica Figura(1.7) $\nabla : [0, 1] \times [0, 1] \longrightarrow [0, 1]$ com

$$x \nabla y = \begin{cases} x & \text{se } y = 0; \\ y & \text{se } x = 0; \\ 1 & \text{caso contrário.} \end{cases}$$

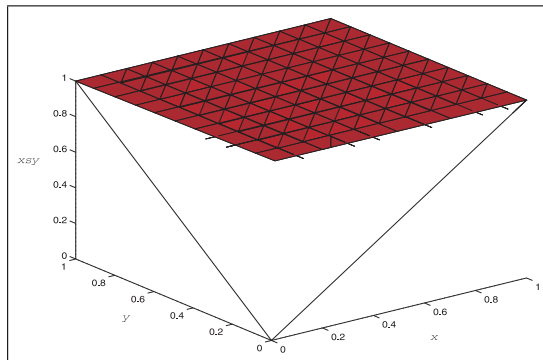


Figura 1.7: Representação da União drástica.[18]

Definição 1.7 Uma norma triangular (*t-norma*) é uma operação binária $\Delta: [0, 1] \times [0, 1] \longrightarrow [0, 1]$ satisfazendo:

- *Comutatividade*: $x \Delta y = y \Delta x$
- *Associatividade*: $x \Delta (y \Delta z) = (x \Delta y) \Delta z$
- *Monotonicidade*: Se $x \leq y$ e $w \leq z$ então $x \Delta w \leq y \Delta z$
- *Condições de Fronteira*: $x \Delta 0 = 0, x \Delta 1 = x$.

Apresentaremos a seguir quatro exemplos de t-norma: Intersecção padrão (operador min), Produto algébrica, Diferença limitada e Intersecção drástica e para cada uma delas mostraremos que vale as condições da Definição 1.7.

1. Intersecção Padrão Figura(1.8) $\Delta: [0, 1] \times [0, 1] \longrightarrow [0, 1]$ com $x \Delta y = \min(x; y)$.

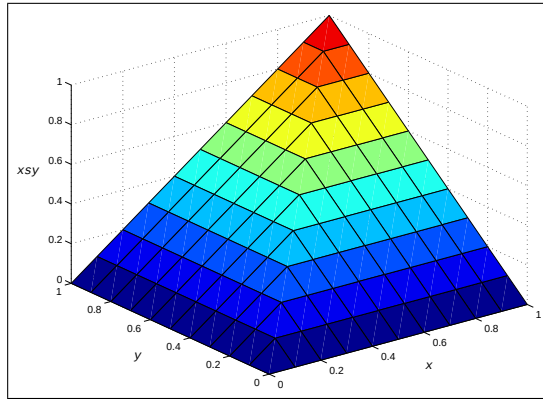


Figura 1.8: Representação da Intersecção padrão.[18]

- (a) Comutatividade: $x \Delta y = y \Delta x$.

De fato, pois $x \Delta y = \min(x, y) = \min(y, x) = y \Delta x$

- (b) Associatividade: $x \Delta (y \Delta z) = (x \Delta y) \Delta z$.

Tem-se que, $x \Delta (y \Delta z) = x \Delta (\min(y, z)) = \min(x, \min(y, z)) = \min(x, y, z)$ e $(x \Delta y) \Delta z = (\min(x, y)) \Delta z = \min(\min(x, y), z) = \min(x, y, z)$.

Daí segue que $x \Delta (y \Delta z) = (x \Delta y) \Delta z$.

- (c) Monotonicidade: Se $x \leq y$ e $w \leq z$ então $x \Delta w \leq y \Delta z$.

Tem-se que, $x \Delta w = \min(x, w)$ e $y \Delta z = \min(y, z)$, por hipótese $x \leq y$ e $w \leq z$, logo $\min(x, w) \leq \min(y, z)$, portanto $x \Delta w \leq y \Delta z$.

- (d) Condições de Fronteira: $x \Delta 0 = 0, x \Delta 1 = x$.

Como $0 \leq x \leq 1$, temos que: $x \Delta 0 = \min(x, 0) = 0$ e $x \Delta 1 = \min(x, 1) = x$.

De (a),(b),(c) e (d) segue pela Definição 1.7 que a intersecção padrão é uma t-norma.

2. Produto Algébrico Figura(1.9) $\Delta: [0, 1] \times [0, 1] \longrightarrow [0, 1]$ com $x \Delta y = xy$.

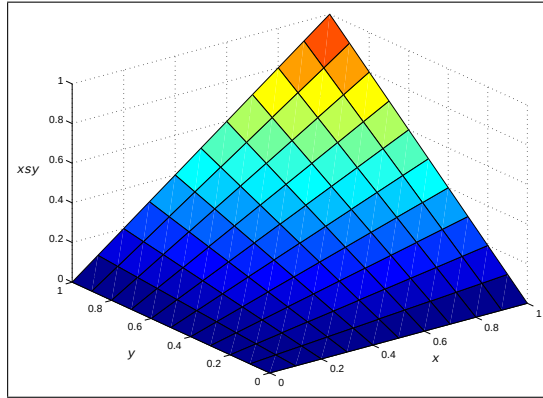


Figura 1.9: Representação do produto algébrico.[18]

(a) Comutatividade: $x \Delta y = y \Delta x$.

De fato, pois $x \Delta y = xy = yx = y \Delta x$.

(b) Associatividade: $x \Delta (y \Delta z) = (x \Delta y) \Delta z$.

Observe que, $x \Delta (y \Delta z) = x \Delta (yz) = xyz$ e $(x \Delta y) \Delta z = (xy) \Delta z = xyz$

Daí segue que $x \Delta (y \Delta z) = (x \Delta y) \Delta z$.

(c) Monotonicidade: Se $x \leq y$ e $w \leq z$ então $x \Delta w \leq y \Delta z$.

Observe que, $x \Delta w = xw$ e $y \Delta z = yz$ e por hipótese $x \leq y$ e $w \leq z$, logo $xw \leq yz$, portanto $x \Delta w \leq y \Delta z$.

(d) Condições de Fronteira: $x \Delta 0 = 0$, $x \Delta 1 = x$.

$x \Delta 0 = x.0 = 0$ e $x \Delta 1 = x.1 = x$.

De (a),(b),(c) e (d) segue pela Definição 1.7 que o produto algébrica é uma t-norma.

3. Diferença Limitada Figura(1.10) $\Delta: [0, 1] \times [0, 1] \longrightarrow [0, 1]$ com $x \Delta y = \max(0; x + y - 1)$.

(a) Comutatividade: $x \Delta y = y \Delta x$.

De fato, pois $x \Delta y = \max(0; x + y - 1) = \max(0; y + x - 1) = y \Delta x$.

(b) Associatividade: $x \Delta (y \Delta z) = (x \Delta y) \Delta z$.

Observe que, $x \Delta (y \Delta z) = x \Delta (\max(0; y + z - 1)) = \max(0; x + y + z - 1 - 1) = \max(0; x + y + z - 2)$ e $(x \Delta y) \Delta z = (\max(0; x + y - 1)) \Delta z = \max(0; x + y + z - 1 - 1) = \max(0; x + y + z - 2)$.

Daí segue que $x \Delta (y \Delta z) = (x \Delta y) \Delta z$.

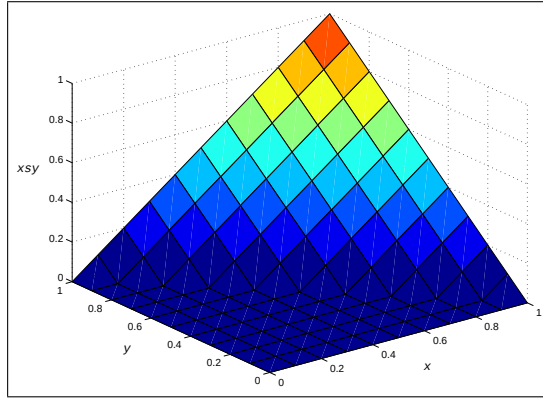


Figura 1.10: Representação da Diferença limitada.[18]

(c) Monotonicidade: Se $x \leq y$ e $w \leq z$ então $x \Delta w \leq y \Delta z$.

Observe que, $x \Delta w = \max(0; x + w - 1)$ e $y \Delta z = \max(0; y + z - 1)$, por hipótese $x \leq y$ e $w \leq z$, logo $x + w - 1 \leq y + z - 1$, portanto $\max(0; x + w - 1) \leq \max(0; y + z - 1)$ segue que $x \Delta w \leq y \Delta z$.

(d) Condições de Fronteira: $x \Delta 0 = 0$, $x \Delta 1 = x$.

Temos que $x \Delta 0 = \max(0; x - 1) = 0$, $x \Delta 1 = \max(0; x + 1 - 1) = \max(0; x) = x$.

De (a),(b),(c) e (d) segue pela Definição 1.7 que a diferença limitada é uma t-norma.

4. Intersecção Drástica Figura(1.11) $\Delta: [0, 1] \times [0, 1] \longrightarrow [0, 1]$ com

$$x \Delta y = \begin{cases} x & \text{se } y = 1; \\ y & \text{se } x = 1; \\ 0 & \text{caso contrário.} \end{cases}$$

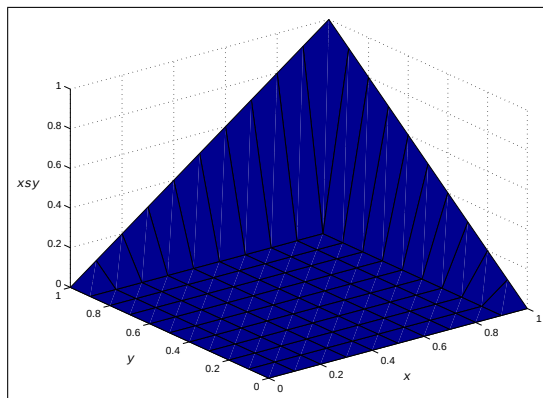


Figura 1.11: Representação da Intersecção drástica.[18]

Capítulo 2

Relações Fuzzy

Veremos neste capítulo que as relações *fuzzy* são, de alguma forma, uma extensão natural das relações matemáticas clássicas.

2.1 Relações Fuzzy

Matematicamente, o conceito de relação é formalizado a partir da teoria de conjuntos. Desta forma, intuitivamente pode-se dizer que a relação será *fuzzy* quando optamos pela teoria dos conjuntos *fuzzy* e será clássica quando optamos pela teoria clássica de conjuntos para conceituar a relação em estudo. Qual dos modelos adotar, entre esses dois, depende muito do fenômeno estudado. Porém, a opção pela teoria de conjuntos *fuzzy* sempre tem maior robustez no sentido de que esta inclui a teoria clássica de conjuntos.

Uma relação clássica segue a função característica da lógica clássica. Sendo assim, uma relação de amizade entre duas pessoas, por exemplo, designadas como “amigos” considera que nas relações humanas ou alguém é seu amigo ou não o é, o que é uma simplificação da realidade. Uma relação de amizade *fuzzy* entre duas pessoas considera o grau de amizade entre elas, sendo assim dois ou mais indivíduos podem se relacionar com diferentes graus de amizade, desde 1.0 (são certamente amigos) até 0.0 (não são amigos). Formalmente, uma relação *fuzzy* R entre duas variáveis, $x \in X$ e $y \in Y$, é definida por uma função que mapeia o par ordenado (x, y) no espaço $X \times Y$ para o seu grau na relação, ou seja, $R : X \times Y \rightarrow [0, 1]$. Esta definição é facilmente generalizada para relações de dimensões superiores.

Definição 2.1 *Uma relação (clássica) $\mathcal{R}$ sobre $U_1 \times U_2 \times \cdots \times U_n$ é qualquer subconjunto (clássico) do produto cartesiano $U_1 \times U_2 \times \cdots \times U_n$. Se o produto cartesiano for formado por apenas dois conjuntos $U_1 \times U_2$, a relação é denominada relação binária. Se $U_1 = U_2 = \cdots = U_n$,*

diz-se que $\mathcal{R}$ é uma relação n -ária sobre U .

Como a relação $\mathcal{R}$ é um subconjunto do produto cartesiano, então ela pode ser representada por sua função característica

$$\chi_{\mathcal{R}} : U_1 \times U_2 \times \cdots \times U_n \rightarrow \{0, 1\},$$

com

$$\chi_{\mathcal{R}}(x_1, x_2, \dots, x_n) = \begin{cases} 1 & \text{se } (x_1, x_2, \dots, x_n) \in \mathcal{R}; \\ 0 & \text{se } (x_1, x_2, \dots, x_n) \notin \mathcal{R}. \end{cases}$$

O conceito matemático de relação *fuzzy* é formalizado a partir do produto cartesiano usual entre conjuntos, estendendo a função característica de uma relação clássica para uma função de pertinência.

Definição 2.2 Uma relação *fuzzy* $\mathcal{R}$, sobre $U_1 \times U_2 \times \cdots \times U_n$, é qualquer subconjunto *fuzzy* do produto cartesiano $U_1 \times U_2 \times \cdots \times U_n$. Se o produto cartesiano for formado por apenas dois conjuntos, $U_1 \times U_2$, a relação é chamada de *fuzzy binária* sobre $U_1 \times U_2$. Assim, uma relação *fuzzy* é definida por uma função de pertinência $\varphi_{\mathcal{R}} : U_1 \times U_2 \times \cdots \times U_n \rightarrow [0, 1]$.

A principal vantagem na opção pela relação *fuzzy* é que a relação clássica indica apenas se há ou não relação entre dois objetos, enquanto uma relação *fuzzy* além de indicar se existe ou não relação, indica também o grau desta relação. Uma noção que será muito importante para este trabalho é o produto cartesiano entre conjuntos *fuzzy*.

Definição 2.3 O produto cartesiano *fuzzy* $A_1 \times A_2 \times \cdots \times A_n$ dos subconjuntos *fuzzy* $A_1, A_2, \dots, A_n$ de $U_1, U_2, \dots, U_n$, é caracterizado pela função de pertinência associada a relação *fuzzy* $\mathcal{R}$ sobre $A_1 \times A_2 \times \cdots \times A_n \subset U_1 \times U_2 \times \cdots \times U_n$:

$$\varphi_{\mathcal{R}}(x_1, x_2, \dots, x_n) = \varphi_{A_1}(x_1) \wedge \varphi_{A_2}(x_2) \wedge \cdots \wedge \varphi_{A_n}(x_n), \quad (2.1)$$

onde $\wedge$ é a t-norma min, isto é, $\varphi_{A_1}(x_1) \wedge \varphi_{A_2}(x_2) = \min(\varphi_{A_1}(x_1), \varphi_{A_2}(x_2))$

e essa será a norma triangular que utilizaremos neste trabalho. O exemplo a seguir ilustra o poder da aplicação do produto cartesiano.

Exemplo 2.1 Consideremos parte da Tabela (1.1) do Exemplo (1.2) que relaciona os diagnósticos de cinco pacientes com dois sintomas, febre e mialgia, conforme Tabela 2.1.

Paciente	Febre: A	Mialgia: B
1	0.7	0.6
2	1.0	1.0
3	0.4	0.2
4	0.5	0.5
5	1.0	0.2

Tabela 2.1: Sintomas

Para diagnosticar um paciente, o médico parte de certas avaliações de sintomas (ou sinais) que são característicos de cada doença. Várias doenças podem apresentar sintomas como febre e mialgia com intensidade e medições diversas. Para a gripe, por exemplo, o paciente apresenta sintomas de “febre” e de “mialgia” com intensidades que, se representadas por subconjuntos fuzzy, devem ter universos distintos. O universo indicador de febre pode ser dado pelas temperaturas possíveis de um indivíduo, enquanto que a mialgia pode ser avaliada pelo número de regiões doloridas.

Para indicar o quanto um indivíduo tem gripe tomamos um grau de pertinência ao conjunto do sintoma febre e ao conjunto mialgia. O paciente 3 da Tabela 2.1, por exemplo, tem uma temperatura x cuja pertinência ao conjunto febre A é $\varphi_A(x) = 0,4$ e tem um valor y de mialgia que faz com que $\varphi_B(y) = 0,2$. O diagnóstico do paciente 3 para a doença gripe é dado por:

$$\text{Paciente 3: } \varphi_{\text{gripe}}(x, y) = \varphi_A(x) \wedge \varphi_B(y) = 0,4 \wedge 0,2 = 0,2.$$

Isto significa que o paciente 3 está no subconjunto fuzzy dos febris com mialgia, tendo grau de pertinência 0,2: que coincide com o seu diagnóstico para a gripe.

Esse número pode dar suporte para, a partir daí, o especialista tomar decisão quanto ao tratamento a ser adotado. É claro que, do ponto de vista teórico, o produto cartesiano clássico também poderia ser adotado para o diagnóstico. Nesse caso, apenas seria indicado gripe (grau um) ou não gripe (grau zero) e, para o exemplo, apenas o paciente 2 da Tabela 2.1 seria considerado gripado.

Na implementação computacional de uma relação fuzzy $\mathcal{R}$, deve-se ter a preocupação de que o código a ser elaborado exercerá o mesmo processo da Função 2.1. Veremos como foi feita essa implementação com mais detalhes na definição de Processamento de dados (seção 4.1.5).

As noção e utilização de produto cartesiano fuzzy ficará mais clara quando introduzirmos o conceito de sistemas baseados em regras fuzzy, que são sistemas compostos de regras da

forma “Se...então...”, pois estas regras podem ser interpretadas como produtos cartesianos de conjuntos *fuzzy*.

2.1.1 Composição de Relações Fuzzy

A composição entre relações é de importância fundamental nas aplicações. Nesta seção apresentaremos apenas a composição mais tradicional em lógica *fuzzy*.

Definição 2.4 *Considere $\mathcal{R}$ e $\mathcal{S}$ duas relações fuzzy binárias em $U \times V$ e $V \times W$, respectivamente. A composição $\mathcal{R} \circ \mathcal{S}$ é uma relação fuzzy binária em $U \times W$, com função de pertinência dada por*

$$\varphi_{\mathcal{R} \circ \mathcal{S}}(u, w) = \max_{v \in V} [\min(\varphi_{\mathcal{R}}(u, v), \varphi_{\mathcal{S}}(v, w))],$$

com $\varphi_{\mathcal{R}}(u, v) = \varphi_U(u) \wedge \varphi_V(v)$ e $\varphi_{\mathcal{S}}(v, w) = \varphi_V(v) \wedge \varphi_W(w)$ sendo $\wedge$ uma t-norma.

Quando os conjuntos U, V e W são finitos, então a forma matricial da relação $\mathcal{R} \circ \mathcal{S}$, dada pela composição max-min, pode ser obtida como uma multiplicação de matrizes substituindo-se o produto pelo mínimo e a soma pelo máximo. De fato suponha que

$$U = \{u_1, u_2, \dots, u_m\}; V = \{v_1, v_2, \dots, v_n\} \text{ e } W = \{w_1, w_2, \dots, w_p\}$$

e que

$$\mathcal{R} = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \dots & r_{mn} \end{pmatrix}_{m \times n} \quad e \quad \mathcal{S} = \begin{pmatrix} s_{11} & s_{12} & \dots & s_{1p} \\ s_{21} & s_{22} & \dots & s_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \dots & s_{np} \end{pmatrix}_{n \times p},$$

de onde temos que

$$r_{ij} = \varphi_{\mathcal{R}}(u_i, v_j) = \varphi_U(u_i) \wedge \varphi_V(v_j) \quad e \quad s_{jk} = \varphi_{\mathcal{S}}(v_j, w_k) = \varphi_V(v_j) \wedge \varphi_W(w_k), \quad (2.2)$$

para $i = 1, \dots, m$, $j = 1, \dots, n$ e $k = 1, \dots, p$.

Assim de acordo com a Definição 2.4, a relação *fuzzy* binária dada pela composição [max-min] tem a forma matricial

$$T = \mathcal{R} \circ \mathcal{S} = \begin{pmatrix} t_{11} & t_{12} & \dots & t_{1p} \\ t_{21} & t_{22} & \dots & t_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ t_{m1} & t_{m2} & \dots & t_{mp} \end{pmatrix}_{m \times p},$$

onde

$$t_{ij} = \max_{1 \leq k \leq n} [\min(\varphi_R(u_i, v_k), \varphi_S(v_k, w_j))] = \max_{1 \leq k \leq n} [\min(r_{ik}, s_{kj})]. \quad (2.3)$$

Exemplo 2.2 (Diagnóstico Médico)

A aplicação que veremos trata de estabelecer diagnóstico para doenças infantis. Este exemplo foi desenvolvido pelas alunas Mariana Fernandes dos Santos Villela e Patrícia Borges dos Santos do Curso de Graduação em Matemática da Universidade Federal de Uberlândia [16], apresentado na 7ª Semana da Matemática da Universidade Federal de Uberlândia [17].

O objetivo é utilizar a composição de relações fuzzy conforme a definição (2.4) em que as relações fuzzy sintomas dos pacientes e das doenças, com esses sinais, ‘captam’ os possíveis diagnósticos dos pacientes.

Para isto, foi preciso consultar um especialista na área. Neste caso foram consultados dois pediatras. A idéia básica é relacionar os sintomas ou sinais de pacientes com as possíveis doenças. Tais doenças são catapora, caxumba, coqueluches e meningite. Considere os seguintes conjuntos universais:

- U_1 = conjuntos dos pacientes do médico 1;
- U_2 = conjuntos dos pacientes do médico 2;
- V = conjunto dos sintomas;
- W = conjunto das doenças.

Foram analisadas as informações de dois médicos diferentes, os quais obteve-se conhecimento de sete pacientes $P_1, P_2, P_3, P_4, P_5, P_6$ e P_7 , com sintomas $s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}, s_{16}, s_{17}$ e s_{18} que apresentaram os diagnósticos d_1, d_2, d_3 e d_4 , onde:

- | | |
|-------------------------------------|---|
| • s_1 = pintas vermelhas no corpo | • s_{10} = infecção das glândulas salivares Febre |
| • s_2 = coceira | • s_{11} = tosse seca |
| • s_3 = febre | • s_{12} = coriza |
| • s_4 = cansaço | • s_{13} = dor muscular |
| • s_5 = cefaléia | • s_{14} = fraqueza |
| • s_6 = perda de apetite | • s_{15} = dor ao mastigar ou engolir |
| • s_7 = rigidez na nuca | • s_{16} = mal estar |
| • s_8 = calafrios | • s_{17} = vômito |
| • s_9 = confusão mental | • s_{18} = dor de garganta |

- $d_1 = \text{catapora}$
- $d_2 = \text{carumba}$
- $d_3 = \text{coqueluche}$
- $d_4 = \text{meningite}$

Neste trabalho será citadas somente as informações do médico 1, as demais podem ser vistas em [8]. Esses dados compõem a base de conhecimentos que são expressos por meio de relações fuzzy. A matriz $\mathcal{R}$, dada abaixo, representa a relação $\mathcal{R}$ em $W \times V$, onde seus valores indicam o grau com que cada sintoma está relacionado com cada doença. Esses valores são as médias aritméticas obtidas através de informações de dois especialistas. As colunas são os sintomas considerados e as linhas são as doenças.

$$\mathcal{R} = \begin{pmatrix} 1 & 1 & 0.4 & 0.45 & 0.5 & 0.4 & 0 & 0.1 & 0 & 0 & 0.2 & 0.3 & 0.05 & 0.2 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.3 & 0.15 & 0.7 & 0.5 & 0 & 0.25 & 0 & 0.8 & 0.1 & 0 & 0.4 & 0.4 & 0.9 & 0.3 & 0.05 & 0.75 \\ 0 & 0 & 0.9 & 0.45 & 0.25 & 0.25 & 0 & 0.15 & 0 & 0 & 1 & 0.55 & 0.1 & 0.1 & 0 & 0.6 & 0.05 & 0 \\ 0.2 & 0 & 0.95 & 0.5 & 0.8 & 0.8 & 1 & 0.75 & 0.4 & 0 & 0 & 0 & 0.3 & 0.1 & 0 & 0.85 & 0.8 & 0 \end{pmatrix}.$$

A matriz $\mathcal{S}$, dada abaixo, representa a relação fuzzy $\mathcal{S}$ em $U_1 \times V$, que indica os graus com que cada sintoma se manifestou nos pacientes, dado pelo especialista 1. As colunas são os sintomas considerados e as linhas são os pacientes. A partir das matrizes das relações fuzzy $\mathcal{R}$ e $\mathcal{S}$ será possível obter o diagnóstico médico de cada paciente, ou seja, o grau de doença para cada paciente, através da Definição 2.4 utilizando a equação 2.3.

$$\mathcal{S} = \begin{pmatrix} 0 & 0 & 0.7 & 0.5 & 0.1 & 0.2 & 0 & 0.5 & 0 & 0 & 1 & 0.5 & 0.1 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0.7 & 0.9 & 0.5 & 0.9 & 0.3 & 0.9 & 0 & 0.5 & 0.1 & 0.6 & 0.5 & 0 & 0.8 & 0.7 & 0 \\ 0 & 0 & 0.5 & 0.3 & 0.8 & 0.7 & 0 & 0.2 & 0 & 1 & 0.5 & 0.2 & 0.3 & 0.5 & 0.9 & 0.7 & 0.3 & 0.8 \\ 1 & 0.8 & 0.9 & 0.3 & 0 & 0.7 & 0 & 0.3 & 0 & 0 & 0 & 0 & 0.2 & 0.3 & 0 & 0.1 & 0 & 0 \\ 1 & 0.5 & 0.9 & 0.2 & 0 & 0.1 & 0 & 0.5 & 0 & 0 & 0 & 0.5 & 0.1 & 0.2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0.1 & 0.1 & 0.1 & 0 & 0.1 & 0 & 0 & 1 & 0.5 & 0.1 & 0.1 & 0 & 0.1 & 0.3 & 0 \\ 0 & 0 & 0.5 & 0.1 & 0.1 & 0.1 & 0 & 0.1 & 0 & 0 & 1.0 & 0.5 & 0.1 & 0.1 & 0 & 0.1 & 0.3 & 0 \end{pmatrix}.$$

Assim a matriz que representa relação fuzzy $\mathcal{D}$ em $W \times U_1$, onde seus valores indicam o grau com que cada paciente está relacionado com cada doença, as linhas são as doenças e as colunas são os pacientes, é obtida através da composição $\mathcal{D} = \mathcal{R} \circ \mathcal{S}^t$, isto é, basta fazer a multiplicação da matriz $\mathcal{R}$ por $\mathcal{S}^t$ (substituindo-se o produto por min e a soma por max).

$$\mathcal{D} = \mathcal{R} \circ \mathcal{S}^t = \begin{pmatrix} 0.45 & 0.5 & 0.6 & 1.0 & 1.0 & 0.3 & 0.45 \\ 0.4 & 0.7 & 0.9 & 0.5 & 0.3 & 0.3 & 0.3 \\ 1.0 & 0.6 & 0.6 & 0.9 & 0.9 & 1.0 & 1.0 \\ 0.7 & 0.9 & 0.8 & 0.95 & 0.9 & 0.3 & 0.5 \end{pmatrix}.$$

Por exemplo, o diagnóstico médico do paciente P_1 , via relação fuzzy $\mathcal{D}$, é facilmente obtido consultando-se a primeira coluna da matriz da relação fuzzy $\mathcal{D}$, pois as colunas e as linhas

representam os pacientes e as doenças, respectivamente. Portanto, notamos que o paciente P_1 , pela teoria aplicada, tem maior possibilidade de estar com coqueluche (d_3). Segundo o especialista, o paciente realmente possuía a respectiva doença.

Note que a resposta da composição é também um conjunto fuzzy, ou seja, a composição nem sempre responde qual doença o paciente possui, porém fornece a possibilidade do paciente no conjunto de doenças dado que ele apresenta uma certa distribuição de possibilidades no conjunto de sintomas. Outra propriedade importante da relação fuzzy é que à medida que tem-se diagnósticos de novos pacientes, estes podem ser incluídos na base de conhecimentos e assim aumentar a capacidade de se obter mais diagnósticos por meio de relações fuzzy, tal como faz o médico.

No próximo capítulo introduziremos as variáveis linguísticas e o sistema baseado em regras fuzzy.

Capítulo 3

Sistema Baseado em Regras Fuzzy

Neste capítulo apresentaremos o Sistema Baseado em Regras *Fuzzy*, sendo tal sistema utilizado para a modelagem de problemas do nosso cotidiano, utilizando a teoria dos conjuntos *fuzzy*.

3.1 Regras e inferência *fuzzy*

Uma regra *fuzzy* é uma sentença da forma ‘Se X é A então Y é B’, onde A e B são conjuntos *fuzzy* em X e Y, respectivamente. Tal regra pode ser interpretada como uma relação *fuzzy* $\mathcal{R}$ entre A e B cuja função de pertinência $\varphi_R(x, y)$ depende de $\varphi_A(x)$ e $\varphi_B(y)$ para cada $(x, y) \in X \times Y$. Neste texto, utiliza-se a função mínimo para essa dependência, ou seja,

$$\varphi_R(x, y) = \varphi_A(x) \wedge \varphi_B(y).$$

Desta forma, $R = A \times B$. Essa foi a modelagem dada por Mamdani para representar a regra ‘Se X é A então Y é B’. Na teoria de raciocínio aproximado, essas sentenças são modeladas por implicações *fuzzy* [2]. Para uma coleção de regras *fuzzy*, usa-se um operador s-conorma para conectá-los, como por exemplo “máximo”.

3.2 Variáveis Linguísticas

As variáveis linguísticas são variáveis que permitem a descrição de informações que estão normalmente disponibilizadas de forma qualitativa, ou seja, são variáveis cujos possíveis valores são palavras ou frases, ao invés de números [14], podendo ser representadas mediante um conjunto *fuzzy*. Estas são expressas qualitativamente através de termos linguísticos, fornecendo um conceito à variável, e quantitativamente por uma função de pertinência. Figura 3.1

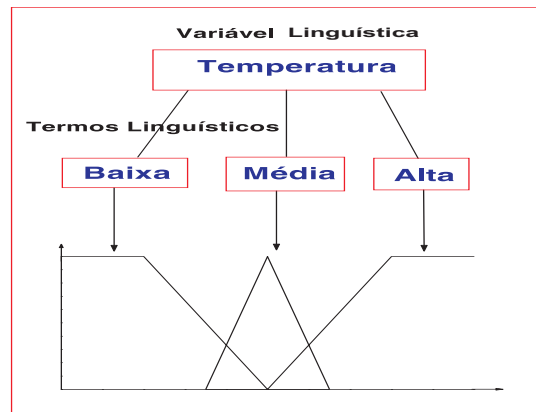


Figura 3.1: Variáveis Linguísticas.[18]

3.2.1 Termos Linguísticos

A cada variável linguística de entrada, devem ser atribuídos termos linguísticos, que representam os estados desta variável. Além disto, deve-se associar um conjunto *fuzzy* a cada termo linguístico de entrada, por meio de uma função de pertinência. Por exemplo, a variável linguística “Temperatura” pode ter o conjunto de termos linguísticos {Baixa, Média, Alta}, sendo que cada termo linguístico representa um conjunto *fuzzy* específico.

3.3 Sistemas Baseados em Regras Fuzzy

No cotidiano, as ações humanas controlam os mais diversos sistemas do mundo real por meio de informações imprecisas. Cada indivíduo funciona como uma “caixa preta”: recebe informações que são interpretadas segundo seus parâmetros e então decide qual atitude tomar. O controle e a execução de tarefas devem seguir uma sequência de “ordens” linguísticas, traduzidas por conjunto de regras, capazes de serem decodificadas pelo controlador.

O exemplo a seguir tem como objetivo ilustrar o comentado acima.

Exemplo 3.1 *Um especialista é capaz de lavar roupas a ponto de deixá-las limpas, segundo o seu conceito de limpeza. O esquema abaixo Figura (3.2) representa, de uma maneira simplificada, as ações de um especialista (controlador humano) na execução da tarefa de lavar roupas.*

Nesse exemplo pode-se observar um possível caminho para a automação de tarefas. As ordens a serem enunciadas por regras, poderiam ser, por exemplo as regras dadas na Tabela 3.1.

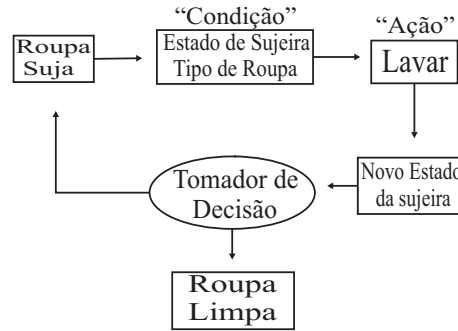


Figura 3.2: Esquema para um sistema de controle humano na tarefa de lavar roupa.

R_1 : Se a roupa é “grossa” e a sujeira é “difícil” então lava-se “muito tempo”.
R_2 : Se a roupa é “grossa” e a sujeira é “fácil” então lava-se “em tempo médio”.
R_3 : Se a roupa é “fina” e a sujeira é “fácil” então lava-se “pouco tempo”.
R_4 : Se a roupa é “fina” e a sujeira é “difícil” então lava-se “pouco tempo”.

Tabela 3.1: Regras de um sistema de automação na lavação de roupas.

Uma tentativa de reproduzir a estratégia de um controlador humano, na execução de suas tarefas, é dada pelos **Controladores Fuzzy**, considerado aqui - a exemplo de tantos outros textos [11] - como um caso típico de um **Sistema Baseado em Regras Fuzzy (SBRF)**, isto é, um sistema que se utiliza da lógica *fuzzy* para produzir saídas para cada entrada *fuzzy*.

Sistemas baseados em regras *fuzzy* (SBRF) contém quatro componentes: um processador de entrada que realiza a fuzzificação dos dados de entrada, uma coleção de regras *fuzzy* chamada de base de regras, uma máquina de inferência *fuzzy* e uma processador de saída que fornece um vetor como saída [8]. Estes componentes estão conectados conforme indicado na figura 3.3.

Supondo $x \in \mathbb{R}^n$ e $y \in \mathbb{R}^m$, temos que um sistema *fuzzy* é uma função de $\mathbb{R}^n$ em $\mathbb{R}^m$ construída de alguma maneira específica. A seguir definiremos os quatro componentes do (SBRF) que indicam um roteiro para a construção desta função.

3.3.1 Processador de Entrada (Fuzzificação)

Neste componente, as entradas do sistema são modeladas em conjuntos *fuzzy* em seus respectivos domínios. É nele que se justifica-se a grande importância de especialistas do fenômeno a ser modelado. Juntamente com os especialistas, as funções de pertinência são formuladas para



Figura 3.3: Sistemas Baseados em Regras *fuzzy*. [7]

cada conjunto *fuzzy* envolvido no processo.

3.3.2 Base de Regras

Este pode ser considerado como um componente do núcleo dos sistemas baseados em regras *fuzzy*. Ele é composto por proposições *fuzzy* e cada uma destas proposições é descrita na forma linguística

$$\begin{array}{ll} \text{Se} & x_1 \text{ é } A_1 \quad e \quad x_2 \text{ é } A_2 \quad e \quad \cdots \quad e \quad x_n \text{ é } A_n \\ \text{Então} & u_1 \text{ é } B_1 \quad e \quad u_2 \text{ é } B_2 \quad e \quad \cdots \quad e \quad u_m \text{ é } B_m \end{array}$$

de acordo com as informações de um especialista. É neste ponto que as variáveis e suas classificações linguísticas são catalogadas e, em seguida, modeladas por conjuntos *fuzzy*, isto é, funções de pertinência. A base de regras descreve relações entre as variáveis linguísticas, para serem utilizadas na máquina de inferência.

3.3.3 Máquina de Inferência *Fuzzy*

Neste componente cada proposição *fuzzy* será “traduzida” matematicamente por meio das técnicas da lógica *fuzzy*. É onde se define quais t-normas, t-conormas e regras de inferência (que podem ser implicações *fuzzy*) serão utilizadas para se obter a relação *fuzzy* que modela a base de regras.

Este componente tem tanta importância quanto o componente da base de regras. Basicamente dele que depende o sucesso do controlador *fuzzy*, já que ele fornecerá a saída (controle) *fuzzy* a ser adotada pelo controlador, a partir de cada entrada *fuzzy*.

3.3.4 Defuzzificação

Na teoria dos conjuntos *fuzzy*, a defuzzificação é um processo que permite representar um conjunto *fuzzy* por um valor crisp (número real)[3], pela Figura (3.3), temos que a base de regras é modelada matematicamente conforme a relação $\mathcal{R}$, a partir dos conjuntos *fuzzy* que a compõe e da lógica *fuzzy* adotada. A função de pertinência de $\mathcal{R}$ é dada por

$$\varphi_{\mathcal{R}}(x, u) = \nabla(\varphi_{R_i}(x, u)), \text{ com } 1 \leq i \leq r, \quad (3.1)$$

onde ∇ é uma t-conorma e R_i é uma relação *fuzzy* obtida da regra i , cuja função de pertinência φ_{R_i} é obtida segundo [3].

De acordo com a literatura, o trabalho pioneiro na automação para realizar e controlar tarefas, baseando-se em lógica *fuzzy*, foi proposto por Mamdani e Assilian [12]. Seus experimentos foram na área de máquina a vapor. Eles se basearam-se no fato que operadores humanos expressam suas estratégias de controle linguisticamente, não de uma forma matematicamente precisa.

A seção a seguir ilustra o método de inferência de Mamdani.

3.4 Método de Inferência de Mamdani

O método de Mamdani é baseado na regra de composição de inferência max-min conforme o procedimento:

Uma regra R_j , da base de regras *fuzzy* é definida pelo produto cartesiano *fuzzy* dos conjuntos *fuzzy* que compõe o antecedente e o consequente da regra. O método de Mamdani agrega as regras através do operador lógico OU, que é modelado pela t-conorma ∇ (máximo) e, em cada regra, o operador lógico E é modelado pela t-norma Δ (mínimo). Veja as regras a seguir:

Regra 1: Se $(x \text{ é } A_1 \text{ e } y \text{ é } B_1)$ então $z \text{ é } C_1$.

Regra 2: Se $(x \text{ é } A_2 \text{ e } y \text{ é } B_2)$ então $z \text{ é } C_2$.

As Figuras 3.4 e 3.5 ilustram como uma saída real z de um sistema do tipo Mamdani é gerada a partir das entradas x e y reais e a regra da composição max-min.

A saída $z \in \mathbb{R}$ é obtida pela defuzzificação do conjunto *fuzzy* de saída $C = C_1 \cup C_2$ da Figura 3.5.

Pode ser visto, através das Figuras 3.4 e 3.5, que a saída do controlador *fuzzy*, dada pelo método de Mamdani, é um subconjunto *fuzzy*, mesmo para o caso de uma entrada crisp. Assim, se for necessário ter um numero real devemos fazer uma defuzzificação do subconjunto *fuzzy* de saída para obter um valor crisp que o represente.

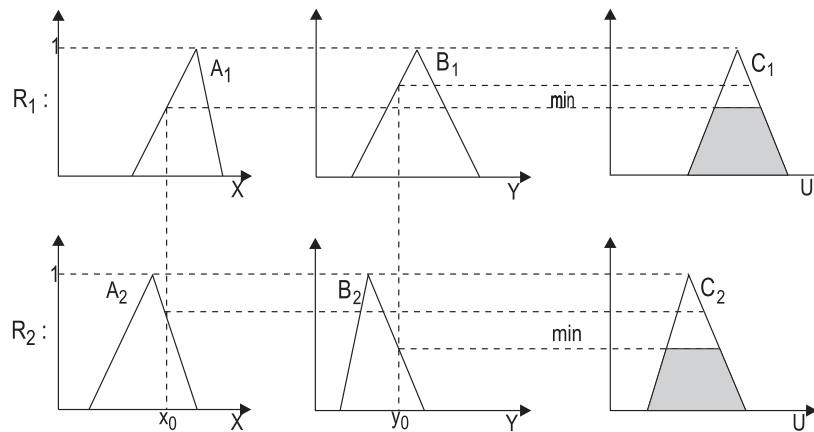


Figura 3.4: Saídas parciais do controlador fuzzy de Mamdani

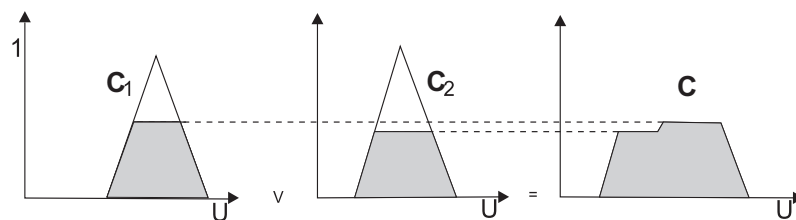


Figura 3.5: Saída final do controlador fuzzy de Mamdani

3.5 Métodos de Defuzzificação

No controlador *fuzzy*, a cada entrada *fuzzy* o módulo de inferência produz uma saída que indica o controle a ser adotado. No entanto se a entrada for um número real, espera-se que a saída correspondente seja também um número real. Porém, isso em geral não ocorre em controladores *fuzzy* pois, mesmo para uma entrada *crisp*, a saída é *fuzzy*. Assim, deve-se indicar um método para defuzzificar a saída e obter um número real que, finalmente, indicará o controle a ser adotado.

São muitos os métodos de defuzzificação que podem ser adotados. A princípio, qualquer número real, que de alguma maneira possa representar razoavelmente o conjunto *fuzzy* C pode ser chamado de defuzzificador de C . Neste trabalho citaremos o mais comum.

3.5.1 Centro de Gravidade ($G(C)$), Centróide ou Centro de Área

Este é o método de defuzzificação que utilizamos neste trabalho, ele é semelhante à média ponderada para a distribuição de dados, com a diferença que os pesos são os valores $\varphi_C(u_i)$, que indicam o grau de compatibilidade do valor u_i com o conceito modelado pelo conjunto *fuzzy* C .

O centro de gravidade dá a média das áreas de todas as figuras que representam os graus

de pertinência de um subconjunto *fuzzy*. Este foi o método, que adotamos neste trabalho, para o componente “Defuzzificação”, entre todos os métodos de defuzzificação ele é o preferido, mesmo sendo talvez o mais complicado. As equações (3.2) e (3.3) referem-se ao domínio discreto e domínio contínuo, respectivamente.

$$G(C) = \frac{\sum_{i=0}^n u_i \varphi_C(u_i)}{\sum_{i=0}^n \varphi_C(u_i)} \quad (3.2)$$

$$G(C) = \frac{\int_R u \varphi_C(u) du}{\int_R \varphi_C(u) du} \quad (3.3)$$

A figura 3.6 mostra o gráfico do defuzzificador $G(C)$.

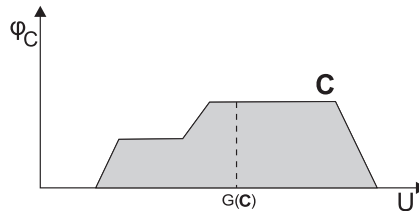


Figura 3.6: Defuzzificador centro de gravidade

Vimos que os controladores *fuzzy* são compostos por quatro componentes: fuzzificação, base de regras, máquina de inferência e defuzzificação. O método de Mamdani é uma caso típico. No entanto, para algumas situações o módulo de defuzzificação pode ser suprimido. Este é o caso do método de inferência de Takagi-Sugeno-Kang que iremos descrever na próxima seção.

3.6 Método de Inferência de Takagi - Sugeno - Kang (TSK)

As diferenças básicas entre o método de inferência de Takagi-Sugeno-Kang (TSK) e o de Mamdani estão na forma de escrever o consequente de cada regra e no procedimento de defuzzificação para se obter a saída geral do sistema. Com o método de TSK, o consequente de cada regra é dado explicitamente por uma função dos valores de entrada desta regra [3].

Como ilustração do método podemos imaginar uma base de regras *fuzzy*, onde cada uma delas tem n entradas $(x_1, x_2, \dots, x_n) \in \mathbb{R}^n$, e uma saída $u \in \mathbb{R}$, conforme o Quadro 3.1, no qual A_{ij} são subconjuntos *fuzzy* de $\mathbb{R}$

$R_1 :$	Se x_1 é A_{11} e x_2 é A_{12} e $\dots$ e x_n é A_{1n} então u é $u_1 = g_1(x_1, x_2, \dots, x_n)$
ou	
$R_2 :$	Se x_1 é A_{21} e x_2 é A_{22} e $\dots$ e x_n é A_{2n} então u é $u_2 = g_2(x_1, x_2, \dots, x_n)$
ou	
$\vdots$	
ou	
$R_r :$	Se x_1 é A_{r1} e x_2 é A_{r2} e $\dots$ e x_n é A_{rn} então u é $u_r = g_r(x_1, x_2, \dots, x_n)$

Quadro 3.1: Base de regras para ilustrar o método de TSK.

A saída geral do método é dada por

$$\begin{aligned}
 u &= f_r(x_1, x_2, \dots, x_n) \\
 &= \frac{\sum_{j=1}^r \omega_j \cdot g_j(x_1, x_2, \dots, x_n)}{\sum_{j=1}^r \omega_j} = \frac{\sum_{j=1}^r \omega_j \cdot u_j}{\sum_{j=1}^r \omega_j},
 \end{aligned} \tag{3.4}$$

onde os pesos ω_j são dados por $\omega_j = \varphi_{A_{j1}}(x_1) \Delta \varphi_{A_{j2}}(x_2) \Delta \dots \Delta \varphi_{A_{jn}}(x_n)$, e Δ é uma t-norma. O peso ω_j corresponde à contribuição da regra R_j para a saída geral. Os casos mais comuns de t-normas são o produto e o mínimo.

Para o caso de duas regras, cada uma com duas variáveis de entrada e uma saída, o método TSK é ilustrado a seguir.

$R_1 :$	“Se x_1 é A_{11} e x_2 é A_{12} então u é $u_1 = g_1(x_1, x_2)$ ”
ou	
$R_2 :$	“Se x_1 é A_{21} e x_2 é A_{22} então u é $u_2 = g_2(x_1, x_2)$ ”

Quadro 3.2: Base de duas regras para o método de TSK.

Supondo que Δ seja a t-norma mínimo, temos como saída geral, representando o controle do sistema para as ações x_1 e x_2 , o valor de u dado pela equação:

$$u = \frac{\omega_1 u_1 + \omega_2 u_2}{\omega_1 + \omega_2} = \frac{\omega_1 g_1(x_1, x_2) + \omega_2 g_2(x_1, x_2)}{\omega_1 + \omega_2} = f_r(x_1, x_2), \tag{3.5}$$

onde $\omega_i = \min[\varphi_{A_{i1}}(x_1), \varphi_{A_{i2}}(x_2)]$ corresponde ao peso da regra R_i na saída geral do processo.

Na literatura, o caso que aparece com maior frequência, devido à sua eficiência e aplicabilidade, é aquele em que os consequentes de cada regra são funções lineares afins, isto é, cada uma das funções g_i tem a forma

$$g_i(x_1, x_2) = a_i x_1 + b_i x_2 + c_i.$$

Este caso é comumente chamado de método de Takagi - Sugeno (TS).

Exemplo 3.2 [3] Considere um controlador fuzzy com duas entradas e uma saída, onde os conjuntos fuzzy envolvidos, A_{ij} , são números fuzzy triangulares e as saídas de cada regra são dadas por funções g_i , lineares afins.

Para cada par de entrada x_0 e y_0 , a Figura 3.7 é uma representação gráfica para a obtenção da saída, a qual representa o controle a ser adotado para tais entradas. Para este exemplo temos a base de regras dadas no Quadro 3.3.

$R_1 :$ “Se x_0 é A_{11} e y_0 é A_{12} então u é $u_1 = g_1(x_0, y_0) = a_1 x_0 + b_1 y_0 + c_1$ ” ou $R_2 :$ “Se x_0 é A_{21} e y_0 é A_{22} então u é $u_2 = g_2(x_0, y_0) = a_2 x_0 + b_2 y_0 + c_2$ ”
--

Quadro 3.3: Base de regras para o Exemplo 3.2.

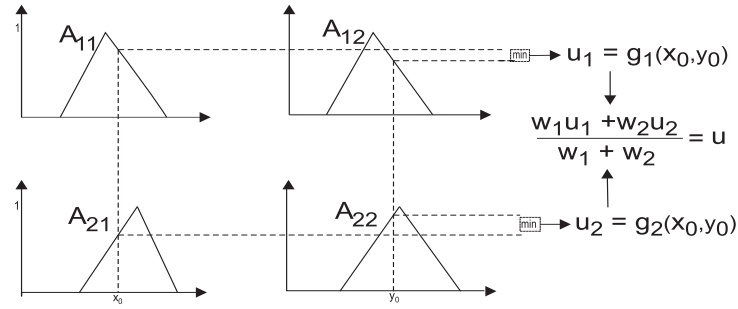


Figura 3.7: Saída do controlador fuzzy TSK para o Exemplo 3.2

Neste caso o controle fuzzy, cuja obtenção gráfica está ilustrada na Figura 3.7, é dado por

$$u = \frac{\omega_1 u_1 + \omega_2 u_2}{\omega_1 + \omega_2} = \frac{\omega_1 g_1(x, y) + \omega_2 g_2(x, y)}{\omega_1 + \omega_2}.$$

Capítulo 4

Algoritmo do Sistema Baseado em Regras Fuzzy (ASBRF)

Neste capítulo apresentaremos o Algoritmo do Sistema Baseado em Regras Fuzzy (ASBRF), que foi desenvolvido com o objetivo de executar os procedimentos referentes ao Sistema Baseado em Regras Fuzzy (SBRF) apresentado no capítulo anterior. Estes procedimentos são bem conhecidos dos usuários do “ferramental *fuzzy*” do *software* MATLAB, denominado *Fuzzy Logic Toolbox*, que é empregado em diversas modelagens matemáticas que utilizam a teoria dos conjuntos fuzzy.

As rotinas que apresentaremos podem ser desenvolvidas em qualquer linguagem de programação, tais como as linguagens C e Fortran. Uma vantagem deste fato consiste na possibilidade de integrar estas rotinas a qualquer código computacional, escrito em uma dada linguagem de programação, especificamente desenvolvido para simular a solução de um problema modelado com a teoria dos conjuntos *fuzzy*, sem a necessidade de se fazer adaptações que permitiriam o acesso ao *Fuzzy Logic Toolbox* do MATLAB.

Uma vantagem do algoritmo que está sendo proposto neste trabalho, em relação aos códigos elaborados no MATLAB, diz respeito à facilidade de se fazer ajustes dinâmicos no SBRF, tais como: i) alteração de parâmetros relacionados à construção das funções de pertinência; isto permitiria a alteração do domínio da função de pertinência e, até mesmo, a alteração do formato desta função; ii) alteração da quantidade de regras do sistema *fuzzy*. Estes ajustes seriam feitos durante a execução de um código computacional, sem a necessidade de interrupções para a inclusão de novas informações. Ajustes como os mencionados anteriormente são necessários, por exemplo, em sistemas parcialmente *fuzzy* (p-*fuzzy*), na construção adequada de uma base de regras que valide a modelagem. Para maiores informações sobre sistemas p-*fuzzy* consulte

[3].

4.1 Estrutura do ASBRF

Para a construção do ASBRF, elaboramos rotinas que, juntas, exercerão os mesmos papéis dos componentes do SBRF. O funcionamento do ASBRF pode ser representado pelo fluxograma Figura 4.1, onde as rotinas estão armazenadas em componentes que conectados formam o ASBRF. A seguir definiremos os componentes do (ASBRF) citando quais as rotinas fazem parte de um determinado componente.

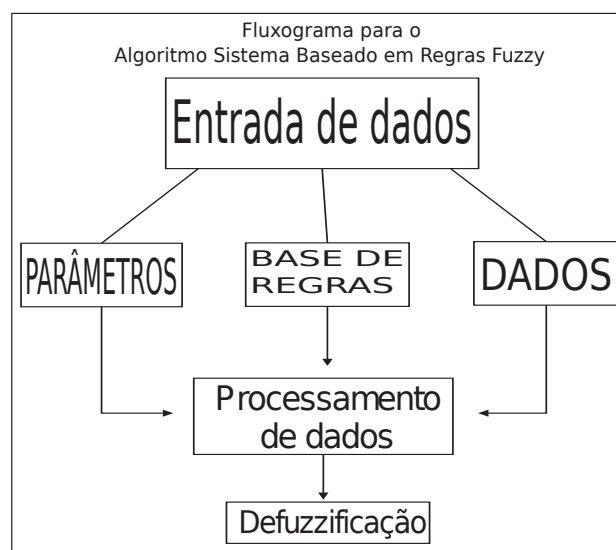


Figura 4.1: Algoritmo Sistema Baseado em Regras Fuzzy (ASBRF)

4.1.1 Entrada de dados

É neste componente que começamos a modelagem matemática de um determinado problema, utilizando a teoria dos conjuntos fuzzy.

A quantidade de variáveis linguísticas de entrada é a primeira informação que necessitamos para a elaboração do algoritmo. Este número será indicado por **NVLE** (Número de Variáveis Linguísticas de Entrada). Além disto, cada variável linguística será qualificada através de termos linguísticos. Os números de termos linguísticos das variáveis de entrada serão armazenados em um vetor denotado por **NTLE** (Número de Termos Linguísticos de Entrada). Assim, a k -ésima coordenada deste vetor, $NTLE(k)$, indicará o número de termos linguísticos da k -ésima variável de entrada.

A variável de saída¹, também é qualificada com termos linguísticos e este número é indicado por **NTLS** (Número de Termos Linguísticos de Saída).

As variáveis **NVLE**, **NTLE** e **NTLS** foram definidas na rotina “main” do ASBRF (ver apêndice 1 na página 77).

4.1.2 Parâmetros

Pela teoria estudada no Capítulo 3, cada termo linguístico deverá ser associado a um conjunto fuzzy, por meio de uma função de pertinência. Neste componente, denominado Parâmetros, serão construídas as funções de pertinência, por meio da rotina “FUNC_PERT” (ver apêndice 1 na página 86). As funções de pertinência, associadas aos termos linguísticos, poderão ser elaboradas com os seguintes formatos:

- Função trapezoidal: Para construirmos a função de pertinência do tipo trapezoidal, definimos a função $f_p = \text{Func_pert}(V, x)$, que depende das variáveis V e x . O vetor de parâmetros $V = (v_1, v_2, v_3, v_4)$, com $v_1 \leq v_2 \leq v_3 \leq v_4$, será utilizado na construção da função de pertinência, a qual será avaliada no ponto x .

$$f_p(x) = \begin{cases} 0 & \text{se } x \leq v_1; \\ \frac{x-v_1}{v_2-v_1} & \text{se } x \in (v_1, v_2]; \\ 1 & \text{se } x \in [v_2, v_3]; \\ -\frac{(x-v_4)}{v_4-v_3} & \text{se } x \in (v_3, v_4]; \\ 0 & \text{se } x \geq v_4, \end{cases}$$

onde v_1, v_2, v_3, v_4 e x pertencem ao conjunto universo de discurso U .

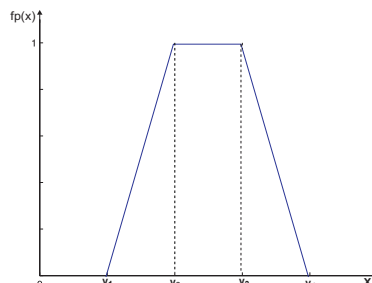


Figura 4.2: Função de pertinência trapezoidal.

¹Neste dissertação, em particular, trabalharemos com a modelagem de problemas com várias entradas e uma única saída

- Função Triangular: Essa função de pertinência é um caso particular da função anterior, bastando o vetor de parâmetros V ter duas de suas coordenadas iguais. Por exemplo, se as coordenadas v_2 e v_3 coincidem então $V = (v_1, v_2, v_2, v_4)$. Assim, a função de pertinência dada por:

$$f_p(x) = \begin{cases} 0 & \text{se } x \leq v_1; \\ \frac{x-v_1}{v_2-v_1} & \text{se } x \in (v_1, v_2]; \\ 1 & \text{se } x \in [v_2, v_3]; \\ -\frac{(x-v_4)}{v_4-v_3} & \text{se } x \in (v_3, v_4]; \\ 0 & \text{se } x \geq v_4. \end{cases}$$

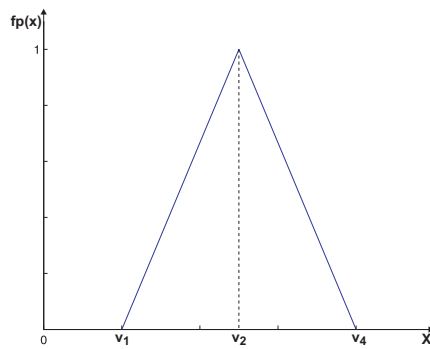


Figura 4.3: Função de pertinência triangular.

- Função Gaussiana: Para este tipo, a função $f_p = func_pert(V, x)$ depende das variáveis V e x . Neste caso, o vetor de parâmetros é dado por $V = (v_1, v_2, v_3)$, em que v_1 e v_2 são os extremos inferior e superior, respectivamente, do intervalo fechado correspondente ao domínio da função f_p ; v_3 é o parâmetro que altera o formato do gráfico desta função (veja a Figura 4.4). A função será avaliada no ponto x .

Para construir a função $f_p(x)$ utilizamos a seguinte função gaussiana [6]

$$H(x) = \begin{cases} \exp\left(\frac{\alpha}{x^2-1}\right) & \text{se } x \in (-1, 1) \\ 0 & \text{se } x \notin (-1, 1) \end{cases},$$

com $\alpha > 0$.

Esta função é definida levando-se em consideração o intervalo $[-1, 1]$. Para construir a função de pertinência f_p levando-se em consideração um intervalo $[a, b]$ qualquer, utiliza-se a seguinte transformação

$$T(x) : [a, b] \rightarrow [-1, 1],$$

dada por

$$T(x) = \left(\frac{2}{b-a}\right)x - \left(\frac{b+a}{b-a}\right),$$

onde $a = v_1, b = v_2$ e $\alpha = v_3$.

A partir desta transformação, a função de pertinência $f_p(x)$ é definida da seguinte forma

$$f_p(x) = c_\alpha H(T(x)) : [a, b] \rightarrow [0, 1].$$

dada por

$$f_p(x) = \begin{cases} c_\alpha \exp\left(\frac{\alpha}{\left(\left(\frac{2}{b-a}\right)x - \left(\frac{b+a}{b-a}\right)\right)^2 - 1}\right) & \text{se } x \in (a, b) \\ 0 & \text{se } x \notin (a, b) \end{cases}, \quad (4.1)$$

onde a constante $c_\alpha = \exp\left(\frac{-\alpha}{\bar{x}^2 - 1}\right)$, sendo $x = \bar{x}$ obtido através da transformação $T(x)$ aplicada no ponto $x_1 = \left(\frac{a+b}{2}\right)$, isto é, $\bar{x} = T(x_1)$. Este valor $x = \bar{x}$ será o ponto de simetria na representação gráfica de $f_p(x)$, e onde a função assumirá o valor 1, isto é, $f_p(\bar{x}) = 1$.

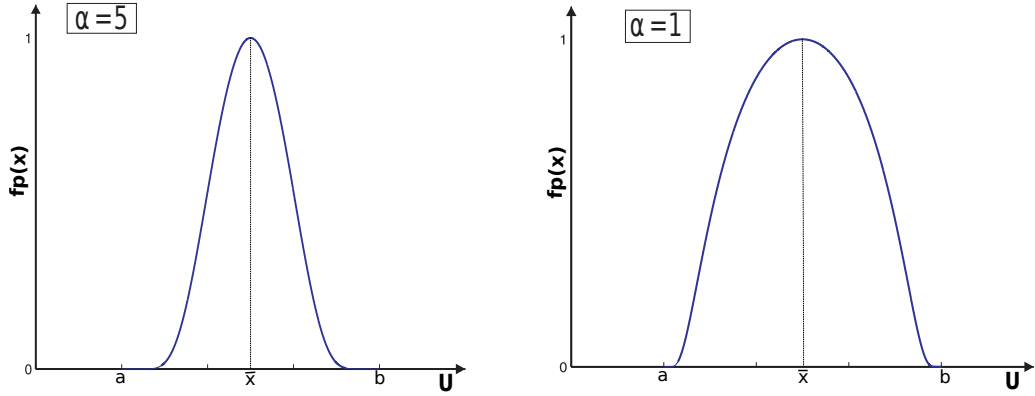


Figura 4.4: Função de pertinência gaussiana para valores α distintos.

Estas são as funções de pertinência que utilizamos neste trabalho. Observe que todos os três tipos de funções dependem de um vetor de parâmetros V . Conhecendo-se a quantidade de variáveis linguísticas de entrada e a quantidade de seus respectivos termos linguísticos é possível construir uma matriz que armazenará todos os vetores de parâmetros associados às funções de pertinência. Esta matriz é denotada por **PTLE** (Parâmetros dos Termos Linguísticos de Entrada) e foi construída na rotina denominada “PARAMETROS_TERMOS_LING_ENTRADA” (ver apêndice 1 na página 82).

A matriz PTLE está dividida por blocos de linhas. Caso existam k variáveis de entrada, o número total de blocos será igual a k . O número total de linhas em cada bloco dependerá do número de termos linguísticos de cada variável de entrada. Assim, cada bloco de linhas estará relacionado a uma determinada variável linguística de entrada e cada linha deste bloco irá corresponder a um vetor de parâmetros, V_i , que dará origem à função de pertinência associada ao termo linguístico i correspondente a essa variável².

Os termos linguísticos serão numerados com números naturais ordenados de forma crescente. Isto permitirá identificar as linhas de um determinado bloco ao seu correspondente termo linguístico. Por exemplo, a variável linguística “Temperatura” da Figura 3.1 é qualificada com os termos linguísticos “Baixa”, “Média” e “Alta”, assim, os termos linguísticos são numerados da seguinte forma: 1 (Baixa), 2 (Média) e 3 (Alta). Neste caso, o bloco da matriz **PTLE**, associado à variável linguística “Temperatura”, seria composto por três linhas, sendo a primeira linha composta pelo vetor de parâmetros V_1 que dará origem à função de pertinência associada ao termo linguístico 1 (Baixa), a segunda linha composta pelo vetor de parâmetros V_2 que dará origem à função de pertinência associada ao termo linguístico 2 (Média) e a terceira linha composta pelo vetor de parâmetros V_3 que dará origem à função de pertinência associada ao termo linguístico 3 (Alta).

O número total de linhas da matriz PTLE, o qual denotamos por n , é igual à soma do número de termos linguísticos de todas as variáveis linguísticas de entrada, isto é,

$$n = \sum_{i=1}^{NVLE} NTLE(i).$$

Seguindo o mesmo procedimento anterior, os vetores de parâmetros que darão origem às funções de pertinência associadas aos termos linguísticos da variável de saída serão armazenados em uma matriz denotada por **PTLS** (Parâmetros dos Termos Linguísticos de Saída). Essa matriz foi construída na rotina “PARAMETROS_TERMOS_LING_SAIDA” (ver apêndice 1 na página 84).

O número de linhas da matriz PTLS, o qual denotamos por m , é igual ao número de termos linguísticos com o qual a variável linguística de saída foi qualificada, isto é, $m = NTLS$.

As três rotinas que foram citadas nesta seção compõem o componente denominado **Parâmetros**, no fluxograma exibido na Figura 4.1.

²O número de colunas dessa matriz é equivalente ao número de coordenadas do vetor de parâmetros V .

4.1.3 Base de Regras

É neste componente que são formuladas todas as relações possíveis entre as variáveis linguísticas. Os termos linguísticos, traduzidos por conjuntos fuzzy, são utilizados para transcrever a base de conhecimentos por meio de uma coleção de regras fuzzy, denominada **base de regras fuzzy**.

Uma base de regras *fuzzy* tem a forma

	R_1 :	“Proposição <i>fuzzy</i> 1”
ou		
	R_2 :	“Proposição <i>fuzzy</i> 2”
		
ou		
	R_r :	“Proposição <i>fuzzy</i> r”

Quadro 4.1.3: Forma geral de uma base de regras *fuzzy*.

Nos sistemas baseados em regras *fuzzy* cada proposição *fuzzy* tem a forma

Se “estado” Então “resposta”

em que cada “estado” e cada “resposta” são valores assumidos por variáveis linguísticas, e esses por sua vez, são modelados por conjuntos *fuzzy*. Os conjuntos *fuzzy* que compõem o “estado” são chamados de *antecedentes*. Por outro lado, os conjuntos *fuzzy* que compõem a “resposta” são chamados *consequentes*.

Na rotina “main”, o ASBRF constrói uma matriz denotada por **Regras**, a qual armazena todas as combinações possíveis entre os termos linguísticos das variáveis linguísticas. Esta parte do algoritmo contém todas as informações dos antecedentes das regras fuzzy. Por exemplo, dadas duas variáveis linguísticas, uma delas qualificada com três termos linguísticos e a outra qualificada com dois termos linguísticos, pode-se construir seis regras fuzzy. Neste caso, a matriz Regras é dada por:

$$Regras = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 2 & 1 \\ 2 & 2 \\ 3 & 1 \\ 3 & 2 \end{pmatrix}.$$

O número total de colunas dessa matriz “Regras” é igual ao número total de variáveis linguísticas de entrada, isto é, igual a $NVLE = 2$, onde a primeira coluna corresponde a primeira variável linguística de entrada e a segunda coluna corresponde a segunda variável linguística de entrada. Cada linha dessa matriz corresponde a uma combinação entre os termos linguísticos (qualificações) das duas variáveis linguísticas de entrada. Por exemplo, na segunda linha temos a combinação entre o termo linguístico representado pelo número **1** da primeira variável linguística de entrada com o termo linguístico representado pelo número **2** da segunda variável linguística de entrada. O número total de linhas dessa matriz, o qual denominamos por r é igual ao número de combinações possíveis entre os termos linguísticos das duas variáveis linguísticas de entrada. Portanto, pelo Princípio Fundamental da Contagem temos que:

$$r = 3 \times 2 = 6.$$

Generalizando o que vimos neste exemplo, o número de colunas para a matriz “Regras”, o qual denotamos por k , será igual ao número de variáveis linguísticas de entrada, isto é, $k = NVLE$, e o número de linhas para essa matriz, o qual denotamos por r , será igual ao número de combinações possíveis entre os termos linguísticos (qualificações) atribuídos às variáveis linguísticas de entrada, isto é,

$$r = \prod_{i=1}^n NTLE(i).$$

Observação 4.1 *O “Sistema Neuro Adaptativo (ANFIS)” é um sistema proposto por Jang (1993)[10] que funciona de modo equivalente aos sistemas de inferência fuzzy. Devido à sua capacidade adaptativa, este sistema é aplicável a uma grande quantidade de áreas de estudos, tais como: classificação de dados e extração de características a partir de modelos. Neste sistema nem sempre a base de regras é composta por todas as combinações possíveis entre os termos linguísticos e sim por um número de regras que melhor se adapte ao sistema. Para a utilização de informações obtidas por este sistema (ANFIS), o algoritmo ASBRF foi adaptado. Criou-se uma rotina denotada por Base de Regras (ver apêndice 4 na página 109), na qual a matriz “Regras” foi construída com o número de regras proposto por este sistema.*

Para completar a formulação do Sistema Baseado em Regras Fuzzy, que é o objetivo do componente denotado por **Base de Regras**, no fluxograma exibido na Figura 4.1, falta informar os consequentes de todas as regras fuzzy. Estes valores serão armazenados em um vetor

coluna denotado por **SBR** (Saída da Base de Regras). Note que a r -ésima coordenada do vetor **SBR**, $\mathbf{SBR}(r)$, corresponde ao consequente da r -ésima regra fuzzy, cujo o antecedente está armazenado na r -ésima linha da matriz “Regras”. O vetor **SBR** foi construído na rotina “SAIDA DA BASE DE REGRAS” (ver apêndice 1 na página 87).

Após a construção da matriz “**Regras**” e do vetor coluna “**SBR**” temos como resultado a matriz “**REGRAS**” que é a concatenação dessas duas anteriores, composta por todas as proposições *fuzzy* na forma “**Se ... então ...**”

Para a utilização do método de inferência de Mamdani, os consequentes são os termos linguísticos atribuídos à variável linguística de saída. Para o programa ASBRF, esses termos linguísticos também estão numerados com números naturais ordenados de forma crescente. Assim, a coordenada r do vetor **SBR** é composta por um número natural p , $p = 1, \dots, NTLS$, que corresponde a um determinado termo linguístico (qualificação) da variável de saída. Por exemplo, dadas duas variáveis linguísticas de entrada, sendo a primeira variável linguística qualificada com três termos linguísticos e a segunda variável linguística qualificada com dois termos linguísticos e uma variável linguística de saída qualificada com dois termos linguísticos, e a matriz **REGRAS** com todas as proposições *fuzzy* sendo definida da seguinte forma:

$$\mathbf{REGRAS} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 2 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 1 & 2 \\ 3 & 2 & 2 \end{pmatrix},$$

a última coluna dessa matriz corresponde ao vetor coluna **SBR**, e as demais colunas correspondem à matriz **Regras**. Observe que a terceira linha da matriz REGRAS corresponde à seguinte proposição *fuzzy*:

Se a primeira variável de entrada for qualificada com o termo linguístico 2 e a segunda variável de entrada for qualificada com o termo linguístico 1;
Então a variável de saída será qualificada com o termo linguístico 1.

Para o método de inferência de Takagi-Sugeno-Kang, o consequente de cada regra, com k valores de entrada, $(x_1, x_2, \dots, x_k)$, é dado explicitamente por uma função $g(x_1, x_2, \dots, x_k)$. Para se utilizar este método, foi elaborada, para este componente, uma rotina denotada por

“SUGENO” (ver apêndice 2 na página 98). As funções (consequentes) anteriores também foram numeradas com números naturais ordenados de forma crescente. Desta forma, a r -ésima coordenada do vetor **SBR** é composta por um número natural, q , que estará associado a uma função (consequente) $g_q(x_1, x_2, \dots, x_k)$, em que $q = 1, \dots, NFS$ e **NFS** (Número de Funções de Sugeno) é o número de funções (consequentes) utilizado na base de regras. Por exemplo, seja a variável linguística “Tempo” qualificada com os termos linguísticos 1 (baixo) e 2 (alto), e as regras *fuzzy* descritas da seguinte forma:

$$R_1 : \text{ Se } t \text{ é baixo } (A_1) \text{ então } g_1(t) = b_1 - a_1 t.$$

$$R_2 : \text{ Se } t \text{ é alto } (A_2) \text{ então } g_2(t) = e^t.$$

Desta forma a matriz “REGRAS” é dada por:

$$\mathbf{REGRAS} = \begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix},$$

sendo a última coluna correspondente ao vetor “SBR” (consequentes), onde os números um e dois estão associados às funções $g_1(t)$ e $g_2(t)$, respectivamente, e a primeira coluna correspondente à matriz “Regras” (antecedentes).

Observação 4.2 *No software MATLAB os consequentes da base de regras, para o método de TSK, são somente funções constantes ou lineares. Na elaboração da rotina “SUGENO”, do ASBRF, os consequentes da base de regras podem ser funções mais gerais. Uma aplicação do ASBRF-SUGENO encontra-se no Capítulo 5 modelo 1 (Decaimento de Fármaco).*

Nesta seção, foram apresentadas as rotinas que compõem o componente **BASE de REGRAS**. Esta parte do algoritmo (ASBRF) tem como objetivo o armazenamento de todas as proposições *fuzzy* na forma “Se ... então ...”, estabelecidas no Sistema Baseado em Regras Fuzzy.

4.1.4 DADOS

Este componente do Algoritmo (ASBRF) contém os valores que serão atribuídos às variáveis linguísticas de entrada. Estes valores são armazenados em um vetor denotado por **DADOS**, de acordo com o fluxograma exibido na Figura 4.1. Este vetor foi construído na rotina “main” e a sua k -ésima coordenada, **DADOS**(k), corresponde ao valor (entrada) atribuído à k -ésima variável linguística de entrada. Por exemplo, se um determinado problema é modelado, utilizando a teoria dos conjuntos *fuzzy*, tendo duas variáveis linguísticas de entrada, sendo a

primeira variável definida no intervalo fechado $[0\ 70]$ e a segunda variável definida no intervalo fechado $[0\ 100]$, então o vetor **DADOS** = $[35\ 50]$ é um dos possíveis vetores que contém os valores atribuídos as duas variáveis de entrada, sendo o valor 35 atribuído a primeira variável e o valor 50 atribuído a segunda variável.

4.1.5 Processamento de dados

Neste componente cada proposição *fuzzy* será “traduzida” matematicamente por meio das técnicas da lógica *fuzzy*. Para isto, foi elaborado, para este componente, a rotina “PROC_DADOS_M” (ver apêndice 1 na página 88). No ASBRF, a chamada desta rotina foi feita como segue.

valor_max_min = PROC_DADOS_M(DADOS, NTLE, NVLE, NTLS, SBR, Regras, PTLE).

Esta rotina depende de quase todas as variáveis construídas nos componentes anteriores, isto mostra que este componente denominado **Processamento de dados**, no fluxograma exibido na Figura 4.1, pode ser considerado como o núcleo para o algoritmo (ASBRF).

Vimos no Capítulo 3 que o método de inferência de Mamdani agrega as regras através do operador lógico OU, que é modelado pela t-conorma ∇ (máximo) e, em cada regra, o operador lógico E é modelado pela t-norma Δ (mínimo).

Como exibido anteriormente, o valor de saída da rotina “PROC_DADOS_M” será armazenado na variável **valor_max_min**. Esta parte do algoritmo executará o mesmo processo executado pelo método de inferência de Mamdani. Lembre-se de que os termos linguísticos da variável linguística de saída estão numerados com números naturais, e que cada linha da matriz “REGRAS” corresponde a uma regra *fuzzy*. Assim, para cada consequente p , $p = 1, \dots, NTLS$, da base de regras será associado um conjunto C_p .

Nos conjuntos C_p são armazenados os valores mínimos obtidos nas regras que têm como consequente o termo linguístico p . Os valores mínimos são obtidos nas regras, através da t-norma Δ (mínimo), da seguinte forma: a numeração das colunas da matriz “Regras” está associada à numeração das variáveis linguísticas de entrada e nas linhas dessa matriz estão armazenadas as informações sobre os termos linguísticos das variáveis de entrada.

Por exemplo, na posição (r, k) da matriz “Regras” está armazenado um número natural i que corresponde a um determinado termo linguístico da variável linguística de entrada k (observe que este número i pode variar de 1 até $NTLE(k)$). À medida que as colunas da matriz “Regras” são percorridas, são efetuados os cálculos da função de pertinência, associada ao termo linguístico $Regras(r, k)$, aplicada na coordenada $DADOS(k)$. Assim, o valor mínimo, dentre

todas as aplicações das funções de pertinência, em uma linha r , é armazenado no conjunto C_p que está associado ao consequente p da regra r .

Para agregar todas as regras, através do operador lógico OU, toma-se o máximo em cada conjunto C_p , tendo no final um vetor denotado por **valor_max_min**. A p -ésima coordenada deste vetor armazena o valor máximo do conjunto C_p . A partir do vetor **valor_max_min**, será construído, na rotina denotada por ‘PERT_DEFUZ’ (ver apêndice 1 na página 90), o subconjunto *fuzzy* C (ver Figura 4.5) que é a saída do algoritmo (ASBRF) dada pelo método de inferência de Mamdani.

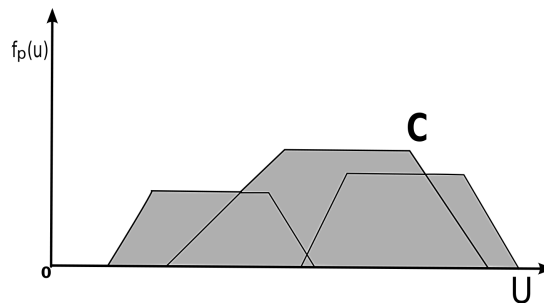


Figura 4.5: Saída final do programa ASBRF para o método de Mamdani.

Um exemplo de utilização da rotina “**PROC_DADOS_M**” encontra-se no final deste capítulo (Vitalidade das Violetas - seção (4.1.9)).

Para o método de inferência de Takagi-Sugeno-Kang, a construção do subconjunto C não é necessária. Como foi visto, no Capítulo 3, os mínimos obtidos na base de regras são utilizados como pesos, ω_i (Equação (3.4)), na saída geral deste método. No método de inferência de TSK, o processamento de dados, no ASBRF, é feito de modo diferente do método de Mamdani. Neste caso, a rotina “**PROC_DADOS_S**” (ver apêndice 2 na página 100) foi definida de maneira análoga à rotina “**PROC_DADOS_M**”. A saída desta função é armazenada no vetor **valor_min** = **PROC_DADOS_S**(Dados, NTLE, NVLE, NFS, SBR, Regras, PTLE), o qual armazenará todos os valores mínimos obtidos na base de regras, ou seja, os pesos ω_i .

Assim, se um determinado problema for modelado com r regras, então o vetor **valor_min** possuirá r coordenadas. A i -ésima coordenada desse vetor armazenará o valor mínimo obtido na i -ésima regra.

Na próxima seção, veremos como será feita a defuzzificação do subconjunto *fuzzy* C , utilizando-se o método de inferência de Mamdani. Apresentaremos, também, a saída geral do método de inferência de Sugeno.

4.1.6 Defuzzificação

Após a obtenção do subconjunto fuzzy de saída, **C**, que foi obtido no componente Processamento de dados do ASBRF, seguindo-se o método de inferência de Mamdani, deve-se executar a defuzzificação do mesmo. Com este intuito, foi elaborada a rotina “**DEFUZZY_M**” (ver apêndice 1 na página 91). No ASBRF, a chamada desta rotina foi feita como segue.

$$\mathbf{DFZ} = \mathbf{DEFUZZY_M}(\mathbf{x_in}, \mathbf{x_fin}, \mathbf{N_SUBINT}, \mathbf{NTLS}, \mathbf{PTLS}, \mathbf{valor_max_min}).$$

As variáveis x_{in} e x_n são os extremos inferior e superior, respectivamente, do intervalo fechado que é o domínio da função de pertinência associado ao subconjunto fuzzy **C**; a variável N_SUBINT é o número de subdivisões do intervalo $[x_{in} \ x_n]$, que será utilizado na regra de integração numérica. As demais variáveis já foram definidas em componentes anteriores.

A rotina “**DEFUZZY_M**” executa o mesmo procedimento realizado no método de defuzzificação conhecido como “Centro de Gravidade ($G(C)$)”, que foi visto na Seção 3.5.1. Este método de defuzzificação é exibido a seguir, para domínios discretos e contínuos:

$$G(C) = \frac{\sum_{i=0}^n u_i f_{pC}(u_i)}{\sum_{i=0}^n f_{pC}(u_i)} \qquad G(C) = \frac{\int_R u f_{pC}(u) du}{\int_R f_{pC}(u) du}.$$

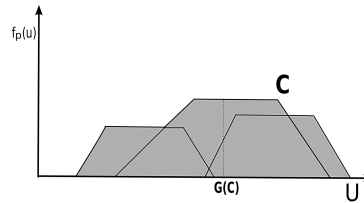


Figura 4.6: Defuzzificador centro de gravidade

As integrais deste método de defuzzificação serão aproximadas pelo método numérico conhecido como “Regra de Simpson” [15] que é definido conforme a expressão

$$\int_a^b f_{pC}(u) du \approx \frac{h}{3} [f_{pC}(u_0) + 4 \sum_{i=1}^{\frac{n}{2}} f_{pC}(u_{2i-1}) + 2 \sum_{i=1}^{\frac{n}{2}-1} f_{pC}(u_{2i}) + f_{pC}(u_n)], \quad (4.2)$$

onde n é um número par positivo e $h = \frac{b-a}{n}$.

A partir desta definição, temos que o centro de gravidade $G(C)$ será expresso como

$$\begin{aligned}
G(C) &= \frac{\int_R u f_{p_C}(u) du}{\int_R f_{p_C}(u) du} \approx \frac{\frac{h}{3} \left[u_0 f_{p_C}(u_0) + 4 \sum_{i=1}^{\frac{n}{2}} u_{2i-1} f_{p_C}(u_{2i-1}) + 2 \sum_{i=1}^{\frac{n}{2}-1} u_{2i} f_{p_C}(u_{2i}) + u_n f_{p_C}(u_n) \right]}{\frac{h}{3} \left[f_{p_C}(u_0) + 4 \sum_{i=1}^{\frac{n}{2}} f_{p_C}(u_{2i-1}) + 2 \sum_{i=1}^{\frac{n}{2}-1} f_{p_C}(u_{2i}) + f_{p_C}(u_n) \right]}, \\
G(C) &\approx \frac{\left[u_0 f_{p_C}(u_0) + 4 \sum_{i=1}^{\frac{n}{2}} u_{2i-1} f_{p_C}(u_{2i-1}) + 2 \sum_{i=1}^{\frac{n}{2}-1} u_{2i} f_{p_C}(u_{2i}) + u_n f_{p_C}(u_n) \right]}{\left[f_{p_C}(u_0) + 4 \sum_{i=1}^{\frac{n}{2}} f_{p_C}(u_{2i-1}) + 2 \sum_{i=1}^{\frac{n}{2}-1} f_{p_C}(u_{2i}) + f_{p_C}(u_n) \right]}. \quad (4.3)
\end{aligned}$$

No Algoritmo (ASBRF), a variável **DFZ** armazena o valor de $G(C)$, que representa o centro de gravidade do subconjunto *fuzzy* **C**.

Para o método de inferência de Takagi-Sugeno-Kang, a saída geral do sistema (defuzzificação) é dada conforme a Equação (3.4) definida no Capítulo 3. No ASBRF, o componente que executará o processo de defuzzificação foi denominado “DEFUZZ_S” (ver apêndice 2 na página 102). Veja a seguir como é feita a chamada desta rotina.

defuzz_Sugeno = DEFUZZ_S(DADOS, NTLE, NVLE, NFS, Saida_Regras, valor_min)

No algoritmo ASBRF, os pesos ω_j são obtidos no processamento de dados e as funções g_j estão definidas na rotina “SUGENO”. Assim, a rotina “DEFUZZ_S” executará o mesmo procedimento da Equação (3.4).

Na próxima seção apresentaremos, para a utilização dos Métodos de inferência de Mamdani e Takagi-Sugeno-Kang, respectivamente, a estrutura computacional do Algoritmo (ASBRF).

4.1.7 Algoritmo (ASBRF - Mamdani) - Estrutura Computacional

Entrada de dados

Entre com o número de variáveis linguísticas de entrada (NVLE);

Entre com o vetor que em cada coordenada conterá a quantidade de termos linguísticos de cada variável linguística de entrada (NTLE);

VLS = 1 (Variável linguística de saída)

Entre com o números de termos linguísticos para a variável linguística de saída (NTLS);

PARÂMETROS

Entre com a matriz contendo os vetores de parâmetros V dos termos linguísticos de entrada (PTLE);

Entre com a matriz contendo os vetores de parâmetros V dos termos linguísticos de saída (PTLS);

BASE DE REGRAS

Construa a matriz “Regras” contendo os antecedentes das regras fuzzy (* Veja o apêndice 1 na página 79);

Entre com o vetor coluna “SBR” contendo os consequentes das regras fuzzy;

REGRAS = [Regras SBR] (acoplamento das matrizes Regras e SBR).

DADOS

Entre com o vetor de dados que em cada coordenada conterá o valor atribuído à respectiva variável linguística de entrada (vetor DADOS);

Processamento de dados

valor_max_min = PROC_DADOS_M(DADOS, NTLE, NVLE, NTLS, SBR, Regras, PTLE) -

Veja o apêndice 1 na página 88

Defuzzificação

DFZ = DEFUZZY_M(x_ini, x_fin, N_SUBINTERV, NTLS, PTLS, valor_max_min) - Veja o apêndice 1 na página 91.

4.1.8 Algoritmo (ASBRF - Sugeno) - Estrutura Computacional

Entrada de dados

Entre com o número de variáveis linguísticas de entrada (NVLE);

Entre com o vetor que em cada coordenada conterá a quantidade de termos linguísticos de cada variável linguística de entrada (NTLE);

Entre com o números de consequentes (funções de Sugeno) (NFS);

PARÂMETROS

Entre com a matriz contendo os vetores de parâmetros V dos termos linguísticos de entrada (PTLE);

BASE DE REGRAS

Construa a matriz “Regras” contendo os antecedentes das regras fuzzy (* Veja o apêndice 2 na página 94);

Entre com o vetor coluna “SBR” contendo os consequentes das regras fuzzy;

REGRAS = [Regras SBR] (acoplamento das matrizes Regras e SBR).

DADOS

Entre com o vetor de dados que em cada coordenada conterá o valor atribuído à respectiva variável linguística de entrada (vetor DADOS);

Processamento de dados

valor_min = PROC_DADOS_S(DADOS, NTLE, NVLE, NFS, SBR, Regras, PTLE) - Veja o apêndice 2 na página 100

Defuzzificação

defuzz_Sugeno = DEFUZZ_S(DADOS, NTLE, NVLE, NFS, Saida_Regras, valor_min) - Veja o apêndice 2 na página 102.

Na próxima seção apresentaremos o exemplo “Vitalidade das Violetas” [8] elaborado pela ex-aluna Flávia Cristina Queiroz do Curso de Graduação em Matemática da Universidade Federal de Uberlândia, mostrando a aplicação do ASBRF utilizando o *software* OCTAVE.

4.1.9 Exemplo - Vitalidade das Violetas

Violeta é um tipo de flor muito apreciada pelos apaixonados por plantas. Possui folhas grandes e folhas miúdas. Para que tenha vida longa, pequenos cuidados diários são necessários. Por exemplo:

- Ser aguada com aproximadamente 33 ml.
- Ser exposta de meia à uma hora ao sol da manhã ou ao da tarde (pois o sol é mais fraco nessas horas).

Para este exemplo foram atribuídas duas variáveis linguísticas de entrada que são: i) Quantidade de Água; ii) Tempo de Exposição ao Sol. Assim, dados os valores da Quantidade de água (ml) e do Tempo de exposição ao Sol (minutos), tem-se como resultado a ‘Vitalidade das Violetas’ (Variável linguística de saída).

As variáveis linguísticas foram qualificadas e quantificadas da seguinte forma:

- Se a Quantidade de água for inferior a $20ml$ (< 20) então o termo linguístico usado para representar esta faixa de valores será *pequena*, as quantidades entre $20ml$ e $38ml$ ($20 - 38$) serão representados pelo termo linguístico *média* e quantidades superiores a $38ml$ ($38ml$) serão representados pelo termo linguístico *grande*.
- Se o tempo de exposição ao Sol (min) for inferior a $30min$ (< 30) então o termo linguístico usado para representar esta faixa de valores será *pequeno*, quando o tempo estiver entre $30min$ e $60min$ o termo linguístico é *médio* e para tempos superiores a $60min$ o termo linguístico é *grande*.
- O domínio da variável de saída ‘Vitalidade das Violetas’ é $[0, 1]$. Com os termos linguísticos: *ruim*, *média* e *boa*.

A Tabela 4.1 apresenta as classificações da vitalidade da violeta como função da quantidade de água A (ml) e o tempo de exposição ao Sol S (min).

Sol (S)	Água (A)	< 20	20 - 38	> 38
< 30		média	boa	ruim
30 - 60		média	boa	ruim
> 60		ruim	média	ruim

Tabela 4.1: Classificações da vitalidade das violetas como função da quantidade de água A (ml) e o tempo de exposição ao Sol S (min).

Para fazer a fuzzificação (Entrada de dados) das informações dadas acima, aplicando o programa **ASBRF**, temos que o número de variáveis linguísticas de entrada é igual a dois, isto é, **NVLE** = 2; e cada uma delas foi qualificada com três termos linguísticos, respectivamente, assim o vetor com a quantidade de termos linguísticos de entrada para cada variável tem a seguinte forma: **NTLE** = [3 3] onde a primeira coordenada corresponde à quantidade de termos linguísticos para a variável linguística “Quantidade de Água” e a segunda coordenada corresponde à quantidade de termos linguísticos para a variável linguística “Tempo de Exposição ao Sol”.

Observação 4.3 *Os termos linguísticos, das duas variáveis linguísticas de entrada e da variável linguística de saída, serão numerados com números naturais ordenados na forma crescente da seguinte forma:*

Quantidade de Água : 1 (*pequena*), 2 (*média*) e 3 (*grande*);

Tempo de Exposição ao Sol : 1 (*pequeno*), 2 (*médio*) e 3 (*grande*);

Vitalidade das Violetas : 1 (*ruim*), 2 (*média*) e 3 (*boa*).

Para cada termo linguístico deve-se associar um conjunto *fuzzy* por meio de uma função de pertinência. Neste caso, utilizaremos para este exemplo funções de pertinência do tipo triangular, sendo os vetores de parâmetros V que darão origem à essas funções, armazenados na matriz **PTLE** que foi construída na rotina PARAMETROS_TERMOS_LING_ENTRADA conforme definição dada na seção 4.1.2.

$$PTLE = \begin{pmatrix} -2 & 0 & 0 & 26 \\ 22 & 33 & 33 & 42 \\ 38 & 68 & 70 & 70 \\ -2 & 0 & 0 & 32 \\ 30 & 42 & 42 & 64 \\ 60 & 98 & 100 & 100 \end{pmatrix},$$

observe que o bloco contendo as três primeiras linhas da matriz PTLE contém os vetores de parâmetros V_1, V_2 e V_3 para a construção das funções de pertinência dos termos linguísticos 1 (*pequena*), 2 (*média*) e 3 (*grande*) da variável linguística de entrada “Quantidade de Água”, e o número total de linhas dessa matriz é igual a seis, que corresponde à soma da quantidade de termos linguísticos da primeira variável com a quantidade de termos linguísticos da segunda variável. Assim a representação gráfica das variáveis linguísticas de entrada será:

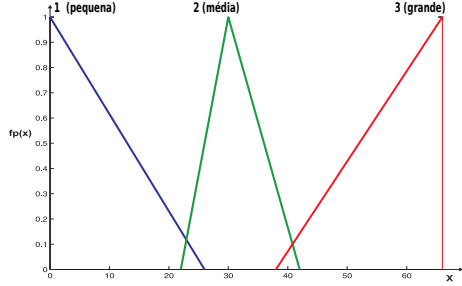


Figura 4.7: Função de pertinência da Quantidade de Água (A)

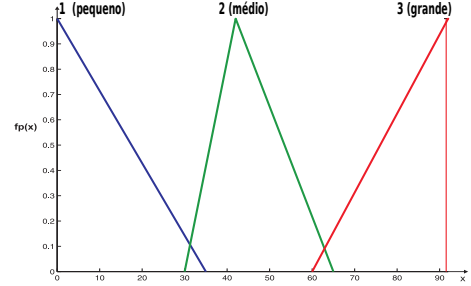


Figura 4.8: Função de pertinência do Tempo de exposição ao Sol (S)

A variável linguística de saída “Vitalidade das Violetas” foi qualificada com três termos linguísticos: 1 (*ruim*), 2 (*média*) e 3 (*boa*). Assim, $\mathbf{NTLS} = 3$.

As funções de pertinência que utilizaremos, para associar os conjuntos *fuzzy* à esses termos linguísticos, são do tipo triangular, e a matriz $\mathbf{PTLS}$ com os vetores de parâmetros V_1, V_2 e V_3 que darão origem às essas funções de pertinência foi construída na rotina `PARAMETROS_TERMOS_LING_SAIDA` conforme a definição dada na seção 4.1.2.

$$PTLS = \begin{pmatrix} -0.1 & 0 & 0 & 0.2 \\ 0.1 & 0.5 & 0.5 & 0.9 \\ 0.8 & 1.0 & 1.0 & 1.0 \end{pmatrix}.$$

A partir desta matriz temos a seguinte representação gráfica para a variável linguística de saída “Vitalidade das Violetas”:

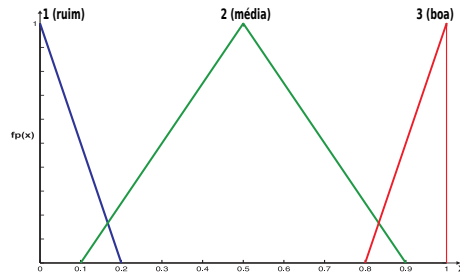


Figura 4.9: Função de pertinência da Vitalidade das Violetas (V).

A partir da Tabela 4.1, a matriz “REGRAS” foi construída com a seguinte forma:

$$\mathbf{REGRAS} = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 2 & 2 \\ 1 & 3 & 1 \\ 2 & 1 & 3 \\ 2 & 2 & 3 \\ 2 & 3 & 2 \\ 3 & 1 & 1 \\ 3 & 2 & 1 \\ 3 & 3 & 1 \end{pmatrix}.$$

A última coluna desta matriz corresponde ao vetor “**SBR**” composto pelos consequentes de cada regra conforme a Tabela 4.1, e as demais colunas correspondem à matriz “**Regras**” composta por todas as combinações entre os termos linguísticos das variáveis linguísticas de entrada.

Observe que a quarta linha da matriz “**REGRAS**” corresponde à seguinte proposição *fuzzy*:

“**Se** a Quantidade de Água for 2 (média) e o Tempo de exposição ao Sol for 1 (pequeno)
Então a Vitalidade das Violetas será 3 (boa)”.

Para simular uma saída do Algoritmo (ASBRF), para este exemplo, atribuímos os seguintes valores para as variáveis de entrada: i) quantidade de água igual a 35 ml; ii) tempo de exposição ao Sol igual a 50 min. Esses valores são armazenados no vetor **DADOS** = [35 50].

Para o Processamento de dados, a cada consequente p , $p = 1, \dots, NTLS$, será associado um conjunto C_p , ou seja, para o consequente 1 (ruim) será associado o conjunto C_1 , para o consequente 2 (média) será associado o conjunto C_2 e para o consequente 3 (boa) será associado o conjunto C_3 . Nestes conjuntos serão armazenados os valores mínimos obtidos na Base de Regras. Por exemplo, a segunda linha da matriz “**REGRAS**” corresponde à seguinte regra *fuzzy*

“**Se** a Quantidade de Água for 1 (pequena) e o Tempo de exposição ao Sol for 2 (médio)
Então a Vitalidade das Violetas será 2 (média)”.

Considere a regra anterior e a função de pertinência f_p associada ao termo linguístico 1 (pequena). Note que esta função de pertinência é determinada através do vetor de parâmetros $V_1 = [-2 \ 0 \ 0 \ 26]$, localizado na primeira linha do primeiro bloco, da matriz PTLE. Lembre-se de que este primeiro bloco está associado à variável linguística “Quantidade de Água”. Agora, considere a primeira coordenada do vetor DADOS, $DADOS(1) = 35$. Observe que $f_p(V_1, 35) = 0$.

Fazendo um procedimento análogo, considerando-se o termo linguístico 2 (médio) da variável linguística “Tempo de exposição ao Sol”, o qual é definido pelo vetor de parâmetros $V_2 = [30 \ 42 \ 42 \ 64]$, obtém-se que $f_p(V_2; DADOS(2)) = f_p(V_2, 50) = 0.636$.

O mínimo entre os dois valores obtidos anteriormente, ou seja, $\min\{0; 0.636\} = 0$ será armazenado no conjunto C_2 , associado ao termo linguístico 2 (média) da variável de saída.

Executando o mesmo procedimento para todas as regras *fuzzy*, a rotina PROC_DADOS_M gera como saída, para o vetor **DADOS** = [35 50], o vetor **valor_max_min** = (0 0 0.6025). Estes valores correspondem aos valores máximos dos conjuntos C_1, C_2 e C_3 .

A partir do vetor **valor_max_min** = (0 0 0.6025), foi construído, na rotina “PERT_DE-FUZ”, o conjunto *fuzzy* **C** dado na Figura 4.10.

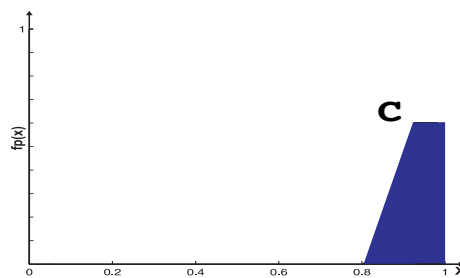


Figura 4.10: Saída final do controlador fuzzy de Mamdani

A saída final do Algoritmo (ASBRF) é obtida através da defuzzificação do conjunto *fuzzy* **C**, que é feita na rotina “DEFUZZY_M”. Esta rotina retorna o valor $DFZ = 0.9296$, que representa o centro de gravidade do conjunto *fuzzy* **C** e nos diz que se a quantidade de água for igual a 35 ml e se o tempo de exposição ao Sol for igual a 50 min então a violeta terá uma boa vitalidade.

Capítulo 5

Aplicações

Para os modelos que apresentaremos aqui, utilizamos o programa ASBRF do Capítulo 4 mostrando assim que o Algoritmo tem a precisão que desejávamos.

5.1 Modelo 1 - Decaimento de Fármaco

O exemplo que apresentaremos foi proposto por Menegotto e Barros [13], está citado como exemplo de aplicações em [3].

Essa é uma aplicação do (SBRF) mostrando o seu potencial de aplicabilidade. Esta aplicação trata-se de um modelo TSK que modela a dinâmica da própria variável de estado de interesse (concentração em função do tempo) a partir de informações de especialistas.

Um problema fundamental em Farmacologia é saber como decai a concentração de uma droga no sangue de um indivíduo. Os fármacos movimentam-se pelo corpo por meio da circulação sanguínea e sabe-se que diferentes grupos de tecidos oferecem maior ou menor resistência na circulação, dependendo da abundância de vasos sanguíneos: grupo de tecidos ricos em vasos sanguíneos (coração, pulmão, fígado, rim e cérebro), grupo muscular de tecidos, com uma quantidade intermediária de vasos e grupos pobres em vasos sanguíneos (gordura, pele e ossos). Esses grupos (ou compartimentos) de tecidos, juntamente com as propriedades físico-química do fármaco, influenciam fortemente na distribuição, retenção e eliminação da substância do organismo [19].

A movimentação de um compartimento para outro modifica continuamente a concentração da droga em cada compartimento até atingir um equilíbrio dinâmico, momento em que não há mais transferência de fármaco entre os compartimentos.

Na prática, a quantidade de compartimentos é indicada a partir do número de pontos alinhados que se destacam no conjunto de dados coletados da concentração do fármaco no

indivíduo, dispostos numa escala tipo monolog. Ver, por exemplo, a Figura 5.1 (b) com duas retas, uma para cada fase: $y_\alpha = b_1 - a_1t$ e $y_\beta = b_2 - a_2t$.

A maioria dos fármacos são analisados por modelos bicompartimentais ou tricompartmentais e o tratamento matemático para prever a concentração no sangue é, via de regra, dado por um sistema de equações diferenciais ordinárias. Nos modelos bicompartimentais tem-se duas etapas de destaque: a fase de distribuição (α - rápida) e a fase de eliminação (β - mais lenta) do fármaco do corpo [9].

A solução de tais sistemas tem a forma

$$C(t) = A_1e^{-\alpha t} + A_2e^{-\beta t}.$$

Este exemplo é baseado nos dados extraídos de [9] que, plotados num plano cartesiano de tempo t (em horas) *versus* o logaritmo da concentração C , indicam claramente tratar-se de um modelo farmacocinético bicompartimental (ver Figura 5.1 (b)). Assim sendo, o modelo matemático tradicional é dado por um sistema de duas equações diferenciais lineares cuja solução (que representa a concentração de fármaco no indivíduo em cada instante t) é

$$C(t) = 1.3e^{-0.173t} + 0.82e^{-0.0092t} \quad (5.1)$$

e as retas de cada fase para esses dados são:

$$y_\alpha = 0.255 - 0.0278t \quad e \quad y_\beta = 0.086 - 0.004t.$$

O interesse agora é reproduzir um modelo *fuzzy* para o sistema bicompartimental levando em conta que as saídas são as retas ajustadas acima. Desta forma, aplicaremos o Algoritmo (ASBRF) utilizando o método de inferência do tipo TSK em que os consequentes são as retas e os antecedentes são “tempo baixo” e “tempo alto”, que aqui adotamos como um número *fuzzy* triangular e um trapezoidal, respectivamente.

Há um valor t^* de maior indecisão nas fases (ver Figura 5.1 (a)):

$$\varphi_{A_1}(t^*) = \varphi_{A_2}(t^*) = 0,5.$$

Esse pode ser dado pela intersecção entre as retas y_α e y_β que, no nosso caso, é $t^* \simeq 14,3h$. Daí o valor $t = 28,6$ que aparece nas fórmulas de φ_{A_1} e φ_{A_2} :

$$\varphi_{A_1}(t) = \begin{cases} 1 - \frac{t}{28,6} & \text{se } 0 \leq t \leq 28,6 \\ 0 & \text{se } t > 28,6 \end{cases}$$

e

$$\varphi_{A_2}(t) = \begin{cases} \frac{t}{28,6} & \text{se } 0 \leq t \leq 28,6 \\ 1 & \text{se } t > 28,6 \end{cases}.$$

As regras *fuzzy* são descritas a seguir:

$$R_1 : \text{ Se } t \text{ é baixo } (A_1) \text{ então } y_\alpha = \log C_\alpha = b_1 - a_1 t \Leftrightarrow C_\alpha = 10^{y_\alpha} \text{ (fase } \alpha)$$

$$R_2 : \text{ Se } t \text{ é alto } (A_2) \text{ então } y_\beta = \log C_\beta = b_2 - a_2 t \Leftrightarrow C_\beta = 10^{y_\beta} \text{ (fase } \beta)$$

Na estrutura computacional do algoritmo (ASBRF), utilizando o método de TSK, temos que: o número de variáveis linguísticas de entrada $NVLE = 1$, o número de termos linguísticos pra essa variável é igual a dois, ou seja, temos o vetor $NTLE = 2$ e a partir das definições das funções de pertinência φ_{A_1} e φ_{A_2} temos a matriz com os parâmetros dos termos linguísticos da seguinte forma

$$PTLE = \begin{pmatrix} 0 & 0 & 0 & 28.6 \\ 0 & 28.6 & 100 & 100 \end{pmatrix},$$

Graficamente a variável linguística de entrada será,

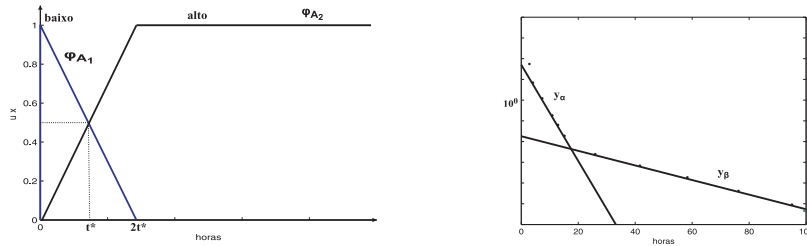


Figura 5.1: (a): Antecedentes. (b) Consequentes y_α e y_β .

sendo a matriz Regras da seguinte forma

$$Regras = \begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix},$$

onde a primeira coluna corresponde a variável de entrada e a segunda coluna correspondente aos consequentes de cada regra, isto é, os números 1 e 2 associados aos consequentes $C_\alpha = g_1(t)$ e $C_\beta = g_2(t)$, respectivamente, onde g_1 e g_2 foram construídas na rotina **SUGENO**.

Assim, pelo método de inferência TSK, fórmula (3.5), a concentração do fármaco no sangue do indivíduo é dada por

$$C(t) = \frac{\varphi_{A_1}(t)10^{y_\alpha} + \varphi_{A_2}(t)10^{y_\beta}}{\varphi_{A_1}(t) + \varphi_{A_2}(t)} \quad (5.2)$$

Tabela 5.1: Resultados do Algoritmo (ASBRF) para a Concentração Sanguínea em relação ao tempo

Amostras	Tempo(horas)	Concentração Estimada (ASBRF)
1	0	1.94984
2	10	1.0435
3	20	0.70007
4	30	0.62230
5	40	0.56754
6	50	0.51761
7	60	0.47206
8	70	0.43053
9	80	0.39264
10	90	0.35810
11	100	0.32659

que a partir das expressões de $y_\alpha, y_\beta, \varphi_{A_1}$ e φ_{A_2} tem-se

$$C(t) = \begin{cases} \left(1 - \frac{t}{28,6}\right) 10^{0,255-0,027t} + \frac{t}{28,6} 10^{-0,086-0,004t} & \text{se } 0 \leq t \leq 28,6 \\ \frac{t}{28,6} 10^{-0,086-0,004t} & \text{se } t > 28,6 \end{cases},$$

Na Tabela 5.1 são apresentados os dados para a concentração sanguínea com as estimativas obtidas pelo Algoritmo (ASBRF).

A representação gráfica, obtida através do Algoritmo, encontra-se na Figura 5.2

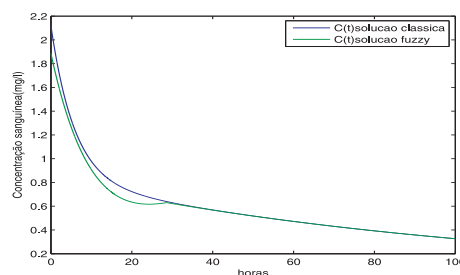


Figura 5.2: Dados, curvas dos modelos determinístico e *fuzzy*

É interessante notar que os gráficos das soluções determinísticas (5.1) e via TSK (5.2) são bem semelhantes e ambas ajustam bem o conjunto de dados.

Se os dados apresentam três retas em destaque, o modelo deveria ser tricompartimental e, conseqüentemente, a base de regras *fuzzy* seria composta por três regras, cujos consequentes

seriam dados por cada uma dessas regras.

A aplicação do ASBRF, utilizando o método de TSK, neste modelo não é substituir o modo como se tem tratado os modelos farmacocinéticos, via equação diferencial. Esta é só mais uma possibilidade de se estudar o decaimento da concentração já que, do ponto de vista prático, aparecem “retas” (a saber y_α e y_β) no estudo de alinhamento dos pontos e o método de inferência TSK é altamente indicado quando os consequentes são retas. Uma vantagem que podemos ressaltar no método *fuzzy* é que este não exige conhecimento algum da teoria de equações diferenciais ordinárias.

Os resultados que obtemos nesta aplicação foram idênticos aos resultados do trabalho original, isto é um resultado satisfatório para a validação do programa ASBRF.

Na utilização do ASBRF nos beneficiamos da vantagem, que este nos oferece, de trabalharmos com funções mais gerais para a saída da base de regras utilizando o método de Takagi-Sugeno, desta forma, para este exemplo, implementamos os consequentes, da base de regras, diretamente na forma exponencial, sem a necessidade de se trabalhar com os consequentes, primeiramente, na forma linear e depois ter que transformar os resultados para a forma exponencial.

5.2 Modelo 2: Modelo *Fuzzy* para a Densidade do Solo

O modelo que apresentaremos agora, foi apresentado através de um minicurso ministrado no CNMAC 2009 e pode ser encontrado na tese de doutorado de Luiz Carlos Benini.

Os dados aqui contidos foram obtidos segundo (Benini, 2007)[4], que para a construção de um modelo *fuzzy* considerando um solo do tipo Latossolo Roxo em que não houve nenhum tipo de preparo, foram utilizados 102 observações compostas de três características do solo na profundidade de 0 a 40 centímetros, que são: a resistência à penetração, representada pelo índice de cone (iCone), em quilopascal (kPa), o teor de água, representada pela umidade do solo, em porcentagem (%), e a textura do solo, representada pela quantidade de argila, em porcentagem (%), entre 30% e 50%. Estas observações foram separadas aleatoriamente em dois arquivos de dados, um constituindo o conjunto de treinamento (78 observações) e o outro o conjunto de teste (24 observações). Foram consideradas três variáveis: duas de entradas, resistência à penetração e teor de água do solo, e uma de saída representando a densidade do solo. Na Tabela 5.2 são mostrados os universos de discurso para cada uma das variáveis.

Para a identificação do modelo adequado para este problema, foram feitas várias simulações (Benini, 2007)[4], utilizando o “Sistema Neuro Adaptativo (ANFIS)”. O modelo ANFIS, pro-

	Variável	Universo de Discurso
entrada	Resistência à penetração (kPa)	[967,4 , 7564,91]
	Teor de água (%)	[20,5 , 32,0]
saída	Densidade do solo (kg.dm-3)	[1,05 , 1,41]

Tabela 5.2: Universos de discurso das variáveis de entrada e saída do sistema *fuzzy* para solo não preparado do tipo III.

posto por Jang (1993)[10], funciona de modo equivalente aos sistemas de inferência *fuzzy*, e suas capacidades adaptativas as fazem aplicáveis a uma grande quantidade de áreas de estudos como, por exemplo, em classificação de dados e extração de características a partir de modelos.

A estrutura ANFIS envolve a seleção de variáveis, a determinação do número de funções de pertinência por variável e a obtenção de um conjunto de regras *fuzzy*. Para se obter um conjunto de regras *fuzzy* Chiu (1994)[5] desenvolveu uma técnica de agrupamento *fuzzy* denominada agrupamento subtrativo, utilizada para particionar o espaço de entrada e saída de um conjunto de dados.

Quando não se conhece “a priori” quantos agrupamentos deve haver para um determinado conjunto de dados, o agrupamento subtrativo é um algoritmo rápido e robusto para saber este número. Ainda, esta técnica permite a localização do centro do agrupamento, sendo as funções de pertinência e as regras obtidas a partir destes centros de agrupamento e, portanto, com estas informações é possível gerar um sistema de inferência *fuzzy* do tipo Takagi-Sugeno, de ordem zero ou um, que modela o comportamento dos dados. Em particular, neste trabalho não apresentaremos o algoritmo do agrupamento subtrativo, se o leitor tiver interesse pode pesquisar (Benini, 2007)[4].

Para a identificação do modelo para o solo não preparado do tipo III, foram encontradas três funções para a variável resistência à penetração e três funções para a variável teor de água, todas do tipo gaussiano.

O modelo identificado para estimar a densidade do solo, é constituído de duas variáveis de entrada, com três conjuntos *fuzzy* associados a cada uma das variáveis, três regras com os antecedentes de cada uma conectados pelo operador “e” e, portanto, três funções lineares, cada qual, representando a parte do consequente da regra *fuzzy*.

Na Tabela 5.2 é dada a caracterização do modelo, obtido através do ANFIS, e na Tabela 5.4 apresenta os parâmetros das funções de pertinência, onde o parâmetro m denota o centro da função e σ a variação dos dados observados com relação a sua média.

Estas funções de pertinência são simétricas e caracterizadas pelos seus centros m e vari-

Modelo de inferência neuro-fuzzy	
Operador “e”	Produto
Número de parâmetros lineares	9
Número de parâmetros não lineares	12
Número total de parâmetros	21
Número de pares de dados treinamento	78
Número de pares de teste	24
Número de regras fuzzy	3

Tabela 5.3: Caracterização do modelo neuro-fuzzy (ANFIS) para solo não preparado do tipo III.

Parâmetros das funções de pertinência (φ)						
Variável	φ_1		φ_2		φ_3	
Entrada	σ	m	σ	m	σ	m
iCone	1399,54	3089,60	1399,54	2518,00	1399,54	5639,60
Umidade	2,435	25,42	2,44	30,29	2,45	20,50

Tabela 5.4: Parâmetros das funções de pertinência para solo não preparado do tipo III

abilidades σ . Para cada função de pertinência pode ser identificado um intervalo para cada um dos espaços da variável de entrada considerada. Assim, tendo como centro do intervalo o valor m e dispersão σ , os intervalos para o conjunto de dados para a variável resistência do solo à penetração são dados por: [967.40 , 7288.22], [967.40 , 6716.62], [1440.98 , 7564.91] e os intervalos que definem os conjuntos *fuzzy* para a variável teor de água (umidade) são dados por: [20.50 , 32.00], [22.97 , 32.00], [20.50 , 27.85].

No trabalho de (Benini, 2007), temos as seguintes representações gráficas, utilizando os dados da Tabela 5.4, para as variáveis linguísticas de entrada (Resistência do Solo e Teor de Água(umidade)).

Na Tabela 5.5 são apresentados os parâmetros das funções lineares dos consequentes, onde os parâmetros C_i , $i = 1, 2, 3$ denotam os coeficientes da função linear do consequente de cada uma das regras do modelo.

Com os intervalos que definem o conjunto *fuzzy* e com os parâmetros apresentados na Tabela 5.5 as regras e as funções lineares que definem as saídas para os consequentes de cada regra, onde x_1 representa a resistência à penetração (índice de cone) e x_2 o teor de água (umidade) no solo, são as seguintes:

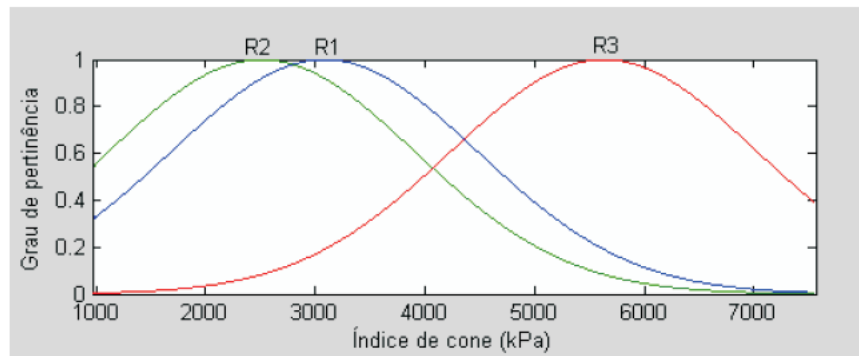


Figura 5.3: Funções de pertinência para a variável de entrada resistência do solo à penetração para o solo não preparado do tipo III.

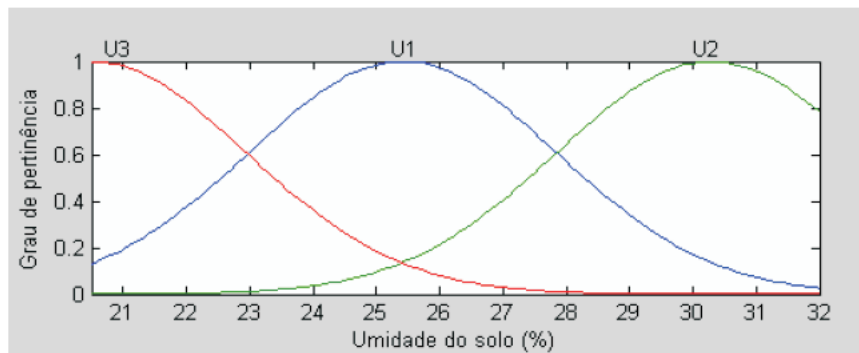


Figura 5.4: Funções de pertinência para a variável de entrada teor de água (umidade) para o solo não preparado do tipo III.

Regra 1. Se $x_1 \in [967.40, 7288.22]$ e $x_2 \in [18.11, 32.00]$

$$\text{Então } y_1 = 0.000045x_1 + 0.031200x_2 + 0.450432$$

Regra 2. Se $x_1 \in [967,40, 6716,62]$ e $x_2 \in [22,97, 32,00]$

$$\text{Então } y_2 = -0.000053x_1 + 0.003389x_2 + 1.6243384$$

Regra 3. Se $x_1 \in [1440,98, 7564,912]$ e $x_2 \in [20,50, 27,85]$

$$\text{Então } y_3 = -0.000023x_1 - 0.006729x_2 + 1.635882$$

Tendo esses dados em mãos, podemos então aplicar o programa ASBRF, utilizando o método de inferência de TSK, construído no Capítulo 4. A mudança que teremos para esta aplicação será na construção da base de regras, no Algoritmo elaboramos o componente “Base de Regras” com o objetivo de construir todas as combinações possíveis entre os termos linguísticos das variáveis linguísticas de entrada. Para o “Sistema Neuro Adaptativo (ANFIS)” nem sempre a base de regras é composta por todas as combinações possíveis entre os termos linguísticos e sim para um número de regras que melhor se adapta ao problema. Neste caso elaboramos

Regras	parâmetros das funções lineares dos consequentes		
	C_1	C_2	C_3
1	0,000045	0,031200	0,450432
2	-0,000053	0,003389	1,243384
3	-0,000023	-0,006729	1,635882

Tabela 5.5: Parâmetros das funções lineares de saída dos consequentes para solo não preparado do tipo III

uma rotina Base de Regras (ver apêndice **42** na página 109), no qual o usuário deve informar o número de regras para o problema (denotado por N_Regras) e poderá construir todas as regras que o modelo (ANFIS) propôs para o problema em questão.

Na aplicação, o número de variáveis linguísticas de entrada é igual a dois, isto é, $NVLE = 2$. Foi proposto para ambas as variáveis linguísticas “Resistência do solo à penetração” e “Teor de água” três termos linguísticos de entrada, assim o vetor $NTLE$ obteve a seguinte forma: $NTLE = [3 \quad 3]$. A partir dos dados obtidos na Tabela 5.4, será construído a representação gráfica para cada variável linguística, utilizando para a construção das funções de pertinência do tipo gaussiano a Função (4.1)(seção 3.2.1) dada por

$$f_p(x) = c_\alpha \exp \left(\frac{\alpha}{\left(\left(\frac{2}{b-a} \right) x - \left(\frac{b+a}{b-a} \right) \right)^2 - 1} \right).$$

Para a construção desta função é necessário informar, o intervalo $[a,b]$ que a define e o parâmetro α que muda sua forma gráfica deixando-a mais fina ou mais larga. No trabalho de (Benini), os intervalos para as funções de pertinência associadas a cada termo linguístico da variável resistência do solo à penetração são dados por: $[967.40, 7288.22]$, $[967.40, 6716.62]$, $[1440.98, 7564.91]$ e os intervalos que definem os conjuntos *fuzzy* para a variável teor de água (umidade) são dados por: $[20.50, 32.00]$, $[22.97, 32.00]$, $[20.50, 27.85]$. A partir destes intervalos e dos dados propostos na Tabela 5.4 (os parâmetros m e σ) será obtido, para cada função de pertinência da forma acima associada a um conjunto *fuzzy*, o intervalo $[a,b]$ e o parâmetro α .

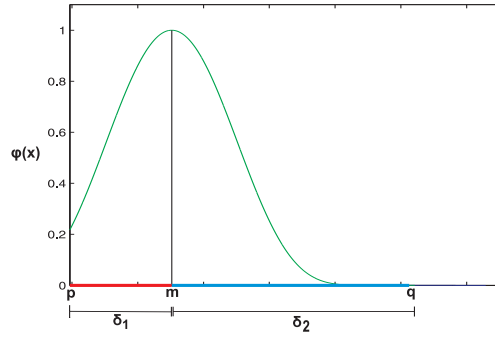
A função de pertinência utilizada no trabalho de (Benini, 2007)[4], foi a seguinte:

$$\varphi(x) = \exp \left(- \frac{1}{2} \frac{(x - m)^2}{\sigma^2} \right), \quad (5.3)$$

onde m é a média e σ é o desvio padrão. Sabe-se, também, que o gráfico desta função é simétrico em relação ao ponto $x = m$ e que os seus pontos de inflexão estão associados ao parâmetro σ .

Para que as funções f_p e φ tenham gráficos similares, os parâmetros $\bar{x}$ e α devem ser ajustados.

Note que $\bar{x} = \frac{a+b}{2}$ é o ponto de simetria da Função 4.1; neste caso, $\bar{x} = m$. Porém, observe que o intervalo $[967.40, 7288.22]$, relacionado à função de pertinência φ_1 , associada ao primeiro termo linguístico da variável resistência do solo à penetração, não é simétrico em relação ao ponto $x = m$ (dado na Tabela 5.4). Para se obter o intervalo correspondente à função f_p , proceda como segue. Sejam, respectivamente, p e q o início e o final do intervalo definido para a função $\varphi(x)$. Considere $\delta_1 = |m - p|$, $\delta_2 = |m - q|$ e $\delta = \max\{\delta_1, \delta_2\}$. Veja a figura abaixo para mais detalhes.



Obtido o valor de δ , basta tomar $a = m - \delta$ e $b = m + \delta$ e assim tem-se o intervalo $[a, b]$ definido de forma que o ponto $x = \bar{x}$ exerça a mesma função do ponto $x = m$.

Definido o intervalo $[a, b]$, resta ainda exibir a correspondência entre os parâmetros α e σ . Para isto, observe que os valores $x = m + \sigma$ e $x = m - \sigma$ são os pontos de inflexão da Função 5.3, isto é, $\varphi''(m \pm \sigma) = 0$. Então, para que as duas funções analisadas aqui possuam uma equivalência no formato de seus gráficos, a distância entre o ponto de simetria e o ponto de inflexão deve ser a mesma em ambos os gráficos. Os pontos de inflexão da Função (4.1) são obtidos a partir da equação $f_p''(x) = 0$.

Note que

$$f_p'(x) = -2c_\alpha c_1 \alpha \frac{(c_1 x - c_2)}{[(c_1 x - c_2)^2 - 1]^2} \exp\left(\frac{\alpha}{(c_1 x - c_2)^2 - 1}\right),$$

sendo $\frac{2}{b-a} = c_1$ e $\frac{b+a}{b-a} = c_2$, portanto

$$f_p''(x) = \frac{2c_\alpha c_1^2 \alpha}{[(c_1 x - c_2)^2 - 1]^4} \left(-[(c_1 x - c_2)^2 - 1]^2 + 4(c_1 x - c_2)^2 [(c_1 x - c_2)^2 - 1] + \right. \\ \left. + 2\alpha (c_1 x - c_2)^2 \right) \exp\left(\frac{\alpha}{(c_1 x - c_2)^2 - 1}\right),$$

de modo que $f_p''(x) = 0$ se, e somente se,

$$\left(-[(c_1 x - c_2)^2 - 1]^2 + 4(c_1 x - c_2)^2 [(c_1 x - c_2)^2 - 1] + 2\alpha (c_1 x - c_2)^2 \right) = 0$$

se, e somente se,

$$3(c_1x - c_2)^4 + 2(c_1x - c_2)^2(\alpha - 1) - 1 = 0.$$

Fazendo $(c_1x - c_2)^2 = y$ na equação acima tem-se que

$$3y^2 + 2y(\alpha - 1) - 1 = 0.$$

As soluções da equação do segundo grau anterior são dadas por

$$y = \frac{-(\alpha - 1) \pm \sqrt{(\alpha - 1)^2 + 3}}{3}.$$

Como $y \geq 0$ e $\alpha > 0$, por hipótese, então a solução procurada tem a seguinte forma:

$$y = \frac{-(\alpha - 1) + \sqrt{(\alpha - 1)^2 + 3}}{3}.$$

Como $(c_1x - c_2)^2 = y$, tem-se que $|c_1x - c_2| = \sqrt{y}$. A partir dessa igualdade, seguem-se dois casos:

1. Sendo $x \geq \frac{c_2}{c_1}$, tem-se que $c_1x - c_2 \geq 0$. Portanto, $c_1x - c_2 = \sqrt{y}$ implica que $x = \frac{c_2}{c_1} + \frac{\sqrt{y}}{c_1}$.
2. Sendo $x < \frac{c_2}{c_1}$, tem-se que $c_1x - c_2 < 0$. Portanto, $-c_1x + c_2 = \sqrt{y}$ implica que $x = \frac{c_2}{c_1} - \frac{\sqrt{y}}{c_1}$.

Primeiramente, observe que $\frac{c_2}{c_1} = \bar{x}$. Assim, os pontos de inflexão da função f_p são dados por $x = \bar{x} + \frac{\sqrt{y}}{c_1}$ e $x = \bar{x} - \frac{\sqrt{y}}{c_1}$.

Como os pontos de inflexão da função $\varphi(x)$ são dados por $x = m + \sigma$ e $x = m - \sigma$, então para se manter a distância entre o ponto de simetria e o ponto de inflexão, no gráfico de ambas as funções, deve-se exigir que

$$\frac{\sqrt{y}}{c_1} = \sigma.$$

Da igualdade anterior, tem-se que $y = \sigma^2 c_1^2$; de onde vem que

$$\frac{-(\alpha - 1) + \sqrt{(\alpha - 1)^2 + 3}}{3} = \sigma^2 c_1^2,$$

obtendo-se para o parâmetro α a seguinte igualdade

$$\alpha = 1 - \frac{3}{2}\sigma^2 c_1^2 + \frac{1}{2\sigma^2 c_1^2}.$$

Sabendo que $\frac{2}{b-a} = c_1$, tem-se, portanto, a seguinte igualdade:

$$\alpha = 1 - \frac{6\sigma^2}{(b-a)^2} + \frac{(b-a)^2}{8\sigma^2}.$$

Determinados o intervalo $[a, b]$ e o parâmetro α , obtemos, a seguir, a matriz *PTLE* com todos os dados necessários para a construção dos gráficos das duas variáveis linguísticas de entrada.

$$PTLE = \begin{pmatrix} -1109.02 & 7288.22 & 1399.54 \\ -1680.62 & 6716.62 & 1399.54 \\ 1440.98 & 9838.22 & 1399.54 \\ 18.50 & 34.00 & 2.43 \\ 22.97 & 37.39 & 2.44 \\ 12.65 & 27.85 & 2.45 \end{pmatrix}.$$

Cada linha desta matriz corresponde a um vetor de parâmetros do tipo $V = (v_1, v_2, v_3)$ que está associado a uma variável linguística k por meio de seu termo linguístico i sendo $k = 1, \dots, NVLE$; $i = 1, \dots, NTLE(k)$. Este vetor V dará origem à função f_p associada a este termo linguístico i . Sendo, respectivamente, v_1 e v_2 o início e o final do intervalo e $v_3 = \sigma$. Por exemplo, na segunda linha da matriz acima localiza-se o vetor de parâmetros $V = (-1680.62, 6716.62, 1399.54)$ que está associado à variável linguística 1 (Resistência do solo à penetração) por meio do seu termo linguístico 2 e na sexta linha desta matriz está o vetor $V = (12.65, 27.85, 2.45)$ que está associado à variável linguística 2 (Teor de água) por meio de seu termo linguístico 3. Concluindo tem-se que o primeiro bloco, desta matriz, contendo as três primeiras linhas corresponde à variável (1) e o segundo bloco contendo as três últimas linhas corresponde à variável (2).

Note que o número total de linhas desta matriz, o qual será denotado por n é igual a soma do número de termos linguísticos das duas variáveis, isto é,

$$n = \sum_{i=1}^{NVLE} NTLE(i) = NTLE(1) + NTLE(2) = 3 + 3 = 6.$$

Utilizando a matriz *PTLE* temos, para a representação gráfica das variáveis linguísticas de entrada, os seguintes gráficos:

A matriz, construída no rotina “Base de Regras”, que contém todas as regras possíveis para este problema tem a seguinte forma

$$Regras = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{pmatrix},$$

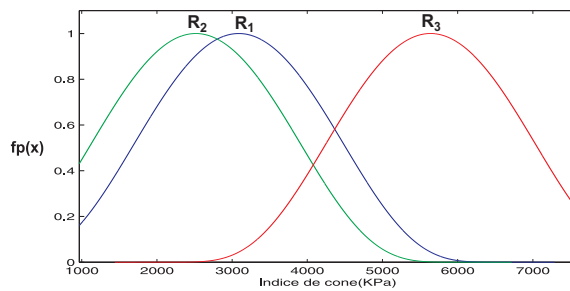


Figura 5.5: Resistência à penetração do Solo

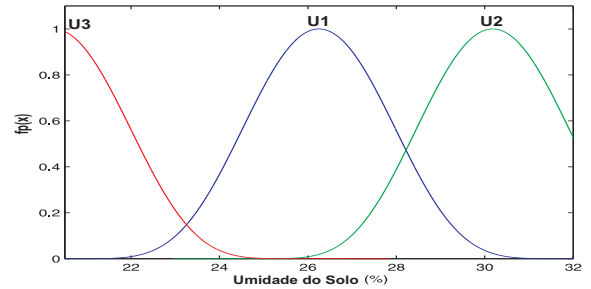


Figura 5.6: Teor de água no Solo (Umidade)

onde a última coluna representa as funções de saída $g_1(x_1, x_2)$, $g_2(x_1, x_2)$ e $g_3(x_1, x_2)$ construídas a partir dos parâmetros da Tabela 5.5 e as demais colunas representam as variáveis de entrada.

Na Tabela 5.6 são apresentados os dados para solo não preparado do tipo III (teor de argila maior que 50%) com as estimativas obtidas pelo modelo encontrado, na tese de doutorado de (Benini, 2007). Nesta tabela serão acrescentados os resultados obtidos na aplicação do Algoritmo.

Observação 5.1 *Para calcularmos o Erro Relativo para os valores obtidos com a aplicação do Algoritmo, utilizamos a seguinte fórmula matemática:*

$$E\% = \frac{|X - x|}{|X|} 100, \quad (5.4)$$

onde X é o valor exato e x é o valor aproximado.

Mas observamos que os valores obtidos por Benini, no cálculo do erro relativo na Tabela 5.6, não satisfazem esta equação. Para fazermos uma comparação equivalente entre as duas aplicações, refazeremos os cálculos, para os dados que Benini obteve em sua aplicação, utilizando a Equação 5.4, tendo os seguintes resultados na Tabela 5.7:

Observe que, os resultados do Modelo comparados com os resultados da Aplicação são bem equivalentes, mostrando assim a validação do Algoritmo.

Tabela 5.6: Resultados do Modelo e do Algoritmo para solo não preparado tipo III (teor de argila maior que 50%)

Amostras	íCone (kPa)	Umidade (%)	Densidade Experimental (kg.dm - 3)	Densidade Estimada (Benini) (kg.dm-3)	Densidade Estimada (Algoritmo) (kg.dm-3)	Erro Relativo (Benini) (%)	Erro Relativo (Algoritmo) (%)
1	3040,00	26,40	1,40	1,36	1,38	3,01	1,42
2	2960,00	26,40	1,40	1,36	1,38	3,21	1,42
3	2270,00	27,00	1,22	1,33	1,34	8,89	9,8
4	2465,00	28,70	1,25	1,29	1,30	2,94	4,00
5	1352,00	30,30	1,21	1,29	1,28	6,55	5,78
6	2180,00	27,00	1,24	1,33	1,34	6,99	8,06
7	4002,00	29,90	1,25	1,22	1,23	2,18	1,60
8	3895,00	27,30	1,30	1,37	1,40	5,15	7,60
9	2933,00	30,00	1,25	1,24	1,23	0,84	1,60
10	3806,00	28,50	1,35	1,30	1,33	3,53	1,48
11	3410,00	28,27	1,33	1,31	1,33	1,62	0
12	3160,00	28,27	1,31	1,31	1,32	0,37	0,76
13	2900,00	28,27	1,29	1,30	1,32	0,97	2,32
14	1976,50	31,90	1,23	1,26	1,24	2,02	0,81
15	4417,50	27,40	1,38	1,38	1,43	0,13	3,62
16	5014,00	27,40	1,38	1,41	1,48	1,80	7,24
17	2094,30	27,40	1,38	1,32	1,33	4,39	3,62
18	3052,60	27,40	1,38	1,34	1,36	3,07	1,44
19	2131,20	22,63	1,39	1,26	1,25	9,11	10,07
20	3378,10	23,85	1,32	1,35	1,34	2,00	1,51
Erro relativo médio (ERM)(Modelo)						3,82	
Erro relativo médio (ERM)(Algoritmo)							3,7

Tabela 5.7: Resultados do Modelo e do Algoritmo para solo não preparado tipo III (teor de argila maior que 50%)

Amostras	íCone (kPa)	Umidade (%)	Densidade Experimental (kg.dm - 3)	Densidade Estimada (Benini) (kg.dm-3)	Densidade Estimada (Algoritmo) (kg.dm-3)	Erro Relativo (Benini) (%)	Erro Relativo (Algoritmo) (%)
1	3040,00	26,40	1,40	1,36	1,38	2,86	1,42
2	2960,00	26,40	1,40	1,36	1,38	2,86	1,42
3	2270,00	27,00	1,22	1,33	1,34	9,02	9,8
4	2465,00	28,70	1,25	1,29	1,30	3,20	4,00
5	1352,00	30,30	1,21	1,29	1,28	6,61	5,78
6	2180,00	27,00	1,24	1,33	1,34	7,26	8,06
7	4002,00	29,90	1,25	1,22	1,23	2,40	1,60
8	3895,00	27,30	1,30	1,37	1,40	5,38	7,60
9	2933,00	30,00	1,25	1,24	1,23	0,80	1,60
10	3806,00	28,50	1,35	1,30	1,33	3,70	1,48
11	3410,00	28,27	1,33	1,31	1,33	1,50	0
12	3160,00	28,27	1,31	1,31	1,32	0,00	0,76
13	2900,00	28,27	1,29	1,30	1,32	0,77	2,32
14	1976,50	31,90	1,23	1,26	1,24	2,44	0,81
15	4417,50	27,40	1,38	1,38	1,43	0,00	3,62
16	5014,00	27,40	1,38	1,41	1,48	2,17	7,24
17	2094,30	27,40	1,38	1,32	1,33	4,35	3,62
18	3052,60	27,40	1,38	1,34	1,36	2,90	1,44
19	2131,20	22,63	1,39	1,26	1,25	9,35	10,07
20	3378,10	23,85	1,32	1,35	1,34	2,27	1,51
Erro relativo médio (ERM)(Modelo)						3,49	
Erro relativo médio (ERM)(Algoritmo)							3,7

5.3 Modelo 3: Conversão de Assintomático para Sintomático. Modelos *Fuzzy* com λ Dependendo do Nível de $CD4^+$ e da Carga Viral

A aplicação que mostraremos agora, se encontra na Tese de Doutorado de Rosana Sueli da Motta Jafelice [7]. Utilizamos este modelo somente para, mais uma vez, validar o Algoritmo.

Quando se analisa a evolução de uma população de indivíduos *HIV*-positivos assintomáticos para a classe dos sintomáticos, muitos fatores estão envolvidos no processo. A taxa de transferência está, pois, sujeita a todos os fatores responsáveis pela mudança de estágios. Alguns destes fatores são mais decisivos que outros, por exemplo, a carga viral do indivíduo e seu nível de $CD4^+$ são fundamentais para predizer seu estado próximo futuro. Este modelo serve para mostrar como é possível estudar um fenômeno, basicamente regido por variáveis inexatas ou incomensuráveis, usando um modelo determinístico com parâmetro subjetivo. Neste caso considera-se a taxa de transferência subjetivamente dependente da carga viral e do nível de $CD4^+$. Assim pode-se expressar tal taxa por um conjunto *fuzzy*, isto é, definindo-a por meio de descrições das variáveis linguísticas.

5.3.1 Informações Médicas sobre HIV

Inicialmente se acreditava que a AIDS tinha um longo período de latência clínica entre a infecção e o desenvolvimento da doença manifesta. Contrária a essa visão, recente pesquisa sobre as contagem de células $CD4^+$ e a replicação viral revela que o estágio intermediário da doença é, na verdade, altamente dinâmico. Essa pesquisa demonstrou, através da análise da meia vida das células, da taxa de replicação viral e da vida média do HIV, que diariamente sobrevive uma quantidade de vírus maior do que as de células $CD4^+$ (o HIV possui uma replicação de 10^{10} vírus/dia e a população de $CD4^+$ é, no máximo, 2×10^9 unidades/dia). Ao longo do tempo essa diferença confere um desequilíbrio em favor do HIV, levando apresentação clínica dos sintomas relacionados à AIDS. Assim, a AIDS é uma consequência dos altos níveis de replicação contínua do HIV em detrimento da menor velocidade de produção de células de defesa, que leva à inutilização e destruição dos linfócitos $CD4^+$, mediadas pelo próprio vírus ou por mecanismos imunológicos.

A contagem de células $CD4^+$ em sangue periférico tem implicações prognósticas na evolução da infecção do HIV, pois é a marca registrada de deficit imunológico e pode ser associada a certos parâmetros clínicos. É a medida de imunocompetência celular mais útil clinicamente

acompanhamento de pacientes infectados pelo HIV e a mais amplamente aceita, embora não seja a única. De maneira didática, pode-se dividir a contagem de células $CD4^+$ por milímetro do sangue periférico em quatro faixas [7]:

- $CD4^+ > 0.5$ células/ml: Estágio da infecção pelo HIV com baixo risco de doença. Neste estágio, há boa resposta às imunizações de rotina e boa confiabilidade nos testes cutâneos de hipersensibilidade tardia como o PPD(Derivado Protéico Purificado). Casos de infecção aguda podem ter estes níveis de $CD4^+$, embora, de modo geral, esses pacientes tenham níveis mais baixos.
- $CD4^+$ entre 0.2 e 0.5 células/ml: Estágio caracterizado por surgimento de sinais e sintomas menores ou alterações constitucionais. Risco moderado de desenvolvimento de doenças oportunistas. Nesta fase podem aparecer candidíase oral, herpes simples recorrente, herpes zóster, tuberculose, leucoplasia pilosa oral, pneumonia bacteriana.
- $CD4^+$ entre 0.05 e 0.2 células/ml: Estágio com alta probabilidade de surgimento de doenças oportunistas como pneumocistose, toxoplasmose de SNC, neurocriptococose, histoplasmose, citomegalovirose localizada. Está associado à síndrome consumptiva, leucoencefalopatia multifocal progressiva, candidíase esofagiana, etc.
- $CD4^+ < 0.05$ células/ml : Estágio com grave comprometimento de resposta imunitária. Alto risco de surgimento de doenças oportunistas como citomegalovirose disseminada, sarcoma de Kaposi, linfoma não-Hodgkin e infecção por microbactérias do complexo Avium-Intracellulare. Alto risco de morte com baixa sobrevida.

A quantificação da carga viral e a contagem de $CD4^+$ são utilizadas para iniciar ou alterar a terapêutica anti-retroviral. Quando não há disponibilidade de quantificação da carga viral pode-se basear na contagem de células $CD4^+$.

Em caso de início ou mudança de terapia anti-retroviral, alguns autores recomendam uma medida de acompanhamento da carga viral após 1 a 2 meses para avaliar o tratamento. Os resultados devem ser interpretados da seguinte maneira:

- Carga viral abaixo de 10.000 cópias de RNA por ml: baixo risco de progressão ou piora da doença.
- Carga viral entre 10.000 e 100.000 cópias de RNA por ml: risco moderado de progressão ou piora da doença.

Situação Clínica	Contagem de $CD4^+$ (células/ml)	Carga Viral (cópias/ml)	Recomendações
Assintomático	Contagem de $CD4^+$ não disponível	Carga viral não disponível	Não tratar
Assintomático	≥ 0.5	Independente da carga viral	Não tratar
Assintomático	$\geq 0.35 < 0.5$	< 3000	Considerar tratamento
		≥ 3000	Considerar tratamento
Assintomático	$\geq 0.2 < 0.35$	Independente de carga viral	Tratamento anti-retroviral
Assintomático	< 0.2	Independente de carga viral	Tratar e iniciar profilaxia para infecções oportunistas
Sintomático	Independente de contagem de $CD4^+$	Independente de carga viral	Tratar e iniciar profilaxia para infecções oportunistas

Tabela 5.8: Recomendações para início da terapia anti-retroviral

- Carga viral acima de 100.000 cópias de RNA por ml: alto risco de progressão ou piora da doença.

Em 2000 o Ministério da Saúde organizou um documento com o título: Recomendações para terapia anti-retroviral em adultos e adolescentes infectados pelo HIV, que contém a Tabela 5.8

5.3.2 O Modelo Clássico

Em 1986, Anderson (et al. [1]) propôs o seguinte modelo para transferência de indivíduos assintomáticos para soropositivos:

$$\begin{cases} \frac{dx}{dt} = -\lambda(t)x & \text{com } x(0) = 1 \\ \frac{dy}{dt} = \lambda(t)x & \text{com } y(0) = 0 \end{cases}, \quad (5.5)$$

onde $\lambda(t)$ é a taxa de transferência de infectados com *HIV*-positivos para indivíduos soropositivos. A variável de estado x indica a proporção de infectados que ainda não têm sintomas declarados de *AIDS* e y indica a proporção de elementos que possuem sintomas claros de *AIDS*.

Em uma primeira aproximação, Anderson propõe que esta taxa deva ser da forma linear

$$\lambda(t) = at, \text{ com } a > 0.$$

Assim a solução do sistema determinístico é dada por:

$$x(t) = e^{-\frac{at^2}{2}} \quad e \quad y(t) = 1 - e^{-\frac{at^2}{2}}.$$

$CD4^+$	V	baixa	média	alta
muito baixo		$z_4 = 1$	$z_4 = 1$	$z_4 = 1$
baixo		$z_3 = 0.65$	$z_4 = 1$	$z_4 = 1$
médio		$z_3 = 0.65$	$z_3 = 0.65$	$z_3 = 0.65$
médio alto		$z_2 = 0.15$	$z_2 = 0.15$	$z_3 = 0.65$
alto		$z_1 = 0$	$z_1 = 0$	$z_3 = 0.65$

Tabela 5.9: Base de regras *fuzzy*

5.3.3 O Modelo *Fuzzy*

Vamos agora considerar a taxa de transferência subjetivamente dependente da *carga viral* v e do *nível de $CD4^+$* c , ou seja,

$$\lambda = \lambda(v, c).$$

Usando uma formulação análoga a do modelo determinístico de Anderson, podemos escrever:

$$\begin{cases} \frac{dx}{dt} = -\lambda(v, c)x \\ x(0) = 1 \end{cases} \quad (5.6)$$

ou sua equação complementar

$$\frac{dy}{dt} = \lambda(v, c)y = k\lambda(v, c)(1 - y) \text{ com } y(0) = 0.$$

Do ponto de vista matemático, podemos pensar em (5.6) como uma família de sistemas de equações diferenciais ordinárias dependendo dos parâmetros. No caso, dependendo de λ , que por sua vez, depende de v e c . Assim, nos parece razoável que o controle de λ , e consequentemente da população y (sintomáticos), possa ser feito a partir de v e c .

Segundo (Jafelice)[7] vamos estimar a taxa de transferência $\lambda = \lambda(v, c)$ baseada nas informações médicas. Adotando-se a base de regras *fuzzy* assumindo como antecedentes a carga viral V e o nível de $CD4^+$, e λ como consequente. Os termos linguísticos para V são *baixa*, *média* e *alta* e para o nível de $CD4^+$ *muito baixo*, *baixo*, *médio*, *médio alto*, e *alto*. Para a taxa de transferência λ os termos linguísticos são *fraca*, *média fraca*, *média* e *forte*.

A Tabela 5.8 relata uma fase importante de transferência de assintomático para sintomático, quando o nível de $CD4^+$ está entre 0.2 e 0.5 cels/ml, assim dividiu-se a contagem de $CD4^+$ em duas faixas: de 0.35 a 0.5 cels/ml não considerar tratamento; e de 0.2 a 0.35 cels/ml considerar tratamento.

Na aplicação utilizamos o método de inferência de Takagi-Sugeno. O número de variáveis linguísticas de entrada é igual a dois, isto é, $NVLE = 2$. Temos, respectivamente, três e cinco

termos linguísticos para as variáveis linguísticas Carga Viral(V) e $CD4^+$, sendo o vetor $NTLE$ da seguinte forma, $NTLE = [3 \ 5]$; A partir da informações médicas e observando a aplicação na tese de doutorado de Jafelice [7], temos a seguinte matriz $PTLE$ com os parâmetros para a construção das funções de pertinência do tipo trapezoidal da Carga viral e do nível de $CD4^+$.

$$PTLE = \begin{pmatrix} -0.1000 & -0.0500 & 0.0200 & 0.1500 \\ 0.0500 & 0.1500 & 0.4500 & 0.5000 \\ 0.4500 & 0.5500 & 1.4500 & 1.5000 \\ -0.1000 & -0.0500 & 0.0100 & 0.1000 \\ 0.0500 & 0.1000 & 0.1500 & 0.2000 \\ 0.1500 & 0.2000 & 0.2800 & 0.3500 \\ 0.2500 & 0.3000 & 0.3500 & 0.5000 \\ 0.4500 & 0.5000 & 1.4500 & 1.5000 \end{pmatrix}.$$

O bloco com as três primeiras linhas representa a variável Carga viral e o segundo bloco com as cinco linhas restantes representa a variável $CD4^+$. Tendo como representação gráfica as Figuras 5.7 e 5.8 dadas abaixo.

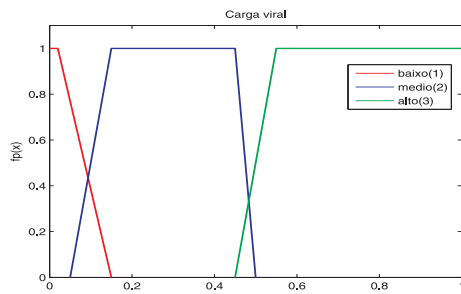


Figura 5.7: Funções de pertinência da Carga viral (V).

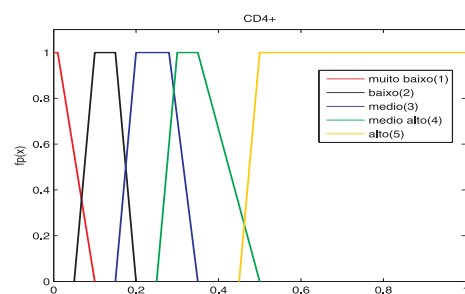


Figura 5.8: Função de pertinência do nível de $CD4^+$.

Na tese de doutorado de Jafelice, dividiu-se os valores da carga viral por 200000 cópias de RNA/ml e com informações médicas construiu-se a Tabela 5.9. Por exemplo: Se V é baixa e $CD4^+$ é muito baixo então λ é forte. A partir disto, construímos a matriz *Regras* com a seguinte

forma:

$$Regras = \begin{pmatrix} 1 & 1 & 4 \\ 1 & 2 & 3 \\ 1 & 3 & 3 \\ 1 & 4 & 2 \\ 1 & 5 & 1 \\ 2 & 1 & 4 \\ 2 & 2 & 4 \\ 2 & 3 & 3 \\ 2 & 4 & 2 \\ 2 & 5 & 2 \\ 3 & 1 & 4 \\ 3 & 2 & 4 \\ 3 & 3 & 3 \\ 3 & 4 & 3 \\ 3 & 5 & 2 \end{pmatrix},$$

onde a última coluna representa as funções de Sugeno $g_1(x_1, x_2)$, $g_2(x_1, x_2)$, $g_3(x_1, x_2)$ e $g_4(x_1, x_2)$ construídas a partir dos parâmetros da Tabela 5.9, para a variável de saída λ , e as demais colunas representando as variáveis linguísticas de entrada.

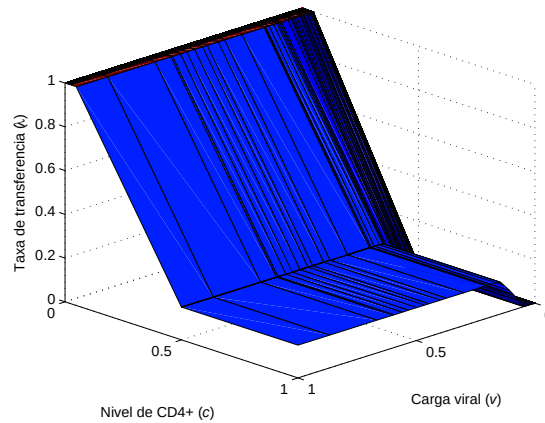


Figura 5.9: Valores da taxa de transferência defuzzificados.

Simulamos 60 valores para a carga viral e nível de $CD4^+$ em um indivíduo HIV-positivo, valores indicados por Jafelice, e determinamos os valores de λ . Com estes resultados, construímos a superfície mostrada na Figura 5.9 e as curvas mostradas nas Figuras 5.10 e 5.11, sendo estas curvas cortes na superfície paralela ao eixo do nível de $CD4^+$ e ao eixo da carga

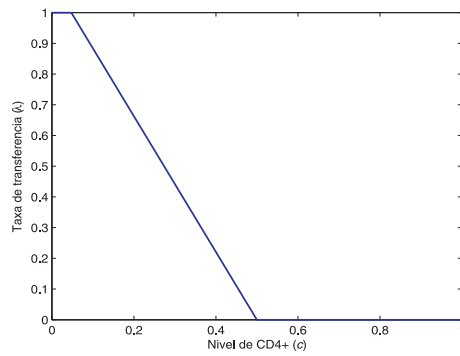


Figura 5.10: λ como função do $CD4^+$ ($v = 0.1$).

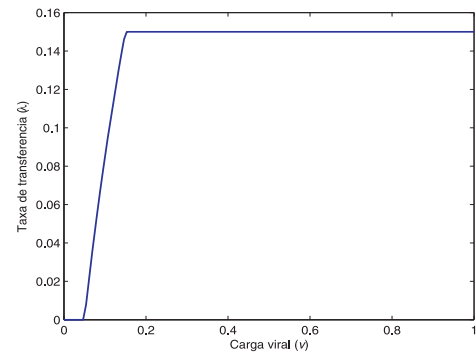


Figura 5.11: λ como função da carga viral ($c = 1$).

viral, respectivamente, as quais serviram para estudos futuros do problema proposto. Estas figuras são idênticas às figuras obtidas no trabalho original e isto é um resultado satisfatório na utilização do algoritmo feito nesta dissertação.

A partir desta aplicação Jafelice obtém outras conclusões em seu trabalho, mas este, não é o nosso objetivo principal. Buscamos aqui mostrar e verificar a utilização do Algoritmo.

Capítulo 6

Conclusão

Neste trabalho foram produzidas rotinas que permitem resolver problemas que utilizam a teoria dos conjuntos *fuzzy*, particularmente aqueles que empregam o sistema baseado em regras *fuzzy*, sem a necessidade de se recorrer a um específico *software*, como o MATLAB, por exemplo. As rotinas *PERT_DEFUZ* e *DEFUZZY*, juntas exercem a mesma função do componente Processador de Saída do **SBRF**; as rotinas *FUNC_PERT* e *PROC_DADOS*, juntas exercem a mesma função do componente Máquina de Inferência do **SBRF**.

No MATLAB, o método de Sugeno só pode ser empregado se os consequentes forem do tipo constante ou linear. Nas rotinas que foram apresentadas nesta dissertação, os consequentes podem ser funções mais gerais. Por exemplo, no Modelo 1 (Decaimento de Fármaco) Capítulo 5, os resultados, na aplicação original, foram obtidos primeiramente por consequentes na forma linear e em seguida transformados para a forma exponencial. Neste trabalho, devido a vantagem de utilizarmos funções mais gerais para os consequentes, os resultados deste modelo foram obtidos diretamente através dos consequentes na forma exponencial.

As rotinas apresentadas neste trabalho podem ser desenvolvidas em qualquer linguagem de programação, tais como as linguagens **C** e Fortran. Uma vantagem deste fato consiste na possibilidade de integrar estas rotinas a qualquer código computacional, escrito em uma dada linguagem de programação, especificamente desenvolvido para simular a solução de um problema modelado com a teoria dos conjuntos fuzzy, sem a necessidade de se fazer adaptações que permitiriam o acesso ao ferramental *fuzzy* (*tool box fuzzy*) do MATLAB.

Com as observações feitas anteriormente, acredita-se que este trabalho vai contribuir com a disseminação da teoria dos conjuntos *fuzzy*, particularmente no que diz respeito ao desenvolvimento de códigos computacionais.

Em trabalhos futuros, pretendemos utilizar os códigos computacionais, que foram desen-

volvidos nesta dissertação, em problemas relacionados à resolução numérica de Equações Diferenciais Parciais. Uma aplicação plausível está associada à modelagem de escoamentos de fluidos em meios porosos, levando-se em conta, principalmente, o estudo dos campos de permeabilidade.

Referências Bibliográficas

- [1] ANDERSON, R. M. et al. *A preliminaire study of the transmission dynamics of the human immunodeficiency virus (HIV), the causitive agent of AIDS*, IMA J. Mathematical Medicine and Biology 3, 1986. p. 229-263.
- [2] BARROS, L.C.; BASSANEZI, R. C. *Tópicos de lógica fuzzy e Biomatemática*, Coleção IMECC - Textos Didáticos,5, 2006.
- [3] BARROS, L. C.; BASSANEZI, R. C. *Tópicos de lógica fuzzy e Biomatemática* - Campinas, SP: UNICAMP/IMECC, 2º ed, 2010.
- [4] BENINI, L. C. *Estimação da densidade de solos utilizando sistemas de inferência fuzzy*. 194 f. Tese (Doutorado em Agronomia) - Faculdade de Ciências Agronômicas, Universidade Estadual Paulista, Botucatu, 2007.
- [5] CHIU, S. L. *Selecting input variables for fuzzy models*, Journal of Intelligent and Fuzzy Systems, 4/4, 1996. p. 243-256.
- [6] FIGUEIREDO, D. G. *Análise I*. Rio de Janeiro, Livros Técnicos e Científicos; Brasília, Ed. Universidade de Brasília, 1975.
- [7] JAFELICE, R. M. *Modelagem Fuzzy para Dinâmica de Transferência de Soropositivos para HIV em Doenças Plenamente Manifesta*, Tese (Doutorado em Engenharia Elétrica) - FEEC, Universidade Estadual de Campinas, Campinas, Brasil, 2003.
- [8] JAFELICE, R. M.; BARROS, L. C.; BASSANEZI, R. C. *Teoria dos Conjuntos Fuzzy com Aplicações*, Uma Publicação da SBMAC - Editora Plêiade, 17, 2005.
- [9] JAMILI, F. *Clinical pharmacokinetics of select classes of drugs: Pharmacokinetic compartments*, Disponível em <<http://www.pharmacy.ualberta.ca/pharm415/pharmaco.htm>>.
- [10] JANG, J. S. *ANFIS: Adaptive-network based fuzzy inference systems*. IEEE Transactions on Systems, Man and Cybernetics, v.23, n.3, 1993. p. 665-685.

- [11] KLIR, G. J.; YUAN, B. *Fuzzy Sets and Fuzzy Logic Theory and Prentice-Hall PTR*. New Jersey, EUA, 1995.
- [12] MAMDANI, E. H.; ASSILIAN, S. *An experiment in linguistic synthesis with a fuzzy logic controller*, Int.J.Man-Machine Studies 7, 1975. p. 1-13.
- [13] MENEGOTTO, J.; BARROS, L. C. *Aplicação da teoria dos conjuntos em modelos farmacocinéticos multicompartimentais*, Tech. report, IMECC - UNICAMP, Sorocaba, 2010
- [14] PEDRYCZ, W.; GOMIDE, F. A. C. *An Introduction to Fuzzy Sets: Analysis and Design*. MIT Press, 1998.
- [15] RUGGIERO, M. A. G.; LOPES, V. L. R. *Cálculo Numérico: aspectos teóricos e computacionais*. 2^a ed. São Paulo: Pearson Education do Brasil, 1996.
- [16] VILLELA, M. F. S.; SANTOS, P. B.; JAFELICE, R. S. M. *Diagnóstico Médico Fuzzy de Doenças Infantis FAMAT em Revista* 9, 2007. p. 329-346.
- [17] VILLELA, M. F. S.; SANTOS, P. B.; JAFELICE, R. S. M. *Diagnóstico Médico Fuzzy de Doenças Infantis*, In *Anais da 7^a Semana da Matemática da Universidade Federal de Uberlândia*, 2007. p. 94-95.
- [18] VILLELA, M. F. S.; ALMEIDA, C. G. *Sistema p-fuzzy e o modelo Malthusiano*. 2008.
- [19] ZANINI, A. C.; OGA, S. *Introdução à farmacologia*, Editora Atheneu, São Paulo, 1989.

Apêndice 1

Código Principal para o Método de Mamdani; Utilizando Funções de Pertinência Trapezoidal ou Triangular

```
% Entrada de dados
% NVLE : Número de variáveis linguísticas das funções de entrada.
clear;
N_SUBINTERV = 10000; % Número de subintervalos que será utilizado para a defuzzificação.
input('VARIÁVEIS LINGUÍSTICAS DAS FUNÇÕES DE ENTRADA - DE ENTER \ n')
NVLE = input('Entre com o número de variáveis linguísticas: ')
caso = input('Voce pode entrar diretamente com o arquivo contendo os termos linguísticos.
Neste caso, DIGITE 1')
if (caso == 1)

    input('Coloque o arquivo na Rotina TERMOS_LING_ENTRADA - De ENTER')

end
%NTLE: Vetor que contém em cada posição  $j$  o número de termos linguísticos para a variável
linguística de entrada  $j$ 
NTLE = TERMOS_LING_ENTRADA(NVLE,caso)
Length_NTLE = length(NTLE);
if (Length_NTLE  $\neq$  NVLE)

    input('ERRO - A dimensão do vetor não confere com o número de variáveis linguísticas')

break;
end
NL_PTLE = 0;
for i = 1:1:Length_NTLE
```

```

NL_PTLE = NL_PTLE + NTLE(i);

end

% PTLE: Matriz que irá armazenar, respectivamente, em cada linha os parâmetros para a cons-
trução das funções de pertinência associadas aos termos linguísticos.

caso2 = input ('Voce pode entrar diretamente com o arquivo contendo os parâmetros dos termos
linguísticos de entrada. Neste caso, Digite 1')

if (caso2 == 1)

    input('Coloque o arquivo na Rotina PARAMETROS_TERMOS_LING_ENTRADA
    - De ENTER')

end

PTLE = PARAMETROS_TERMOS_LING_ENTRADA(NVLE, NTLE, caso2)

if (size(PTLE,1) ≠ NL_PTLE)

    input('ERRO - A dimensão da matriz não confere com o número total de termos
    linguísticos')

break;

end

% Entrada de dados

% VLS : Variável linguística de saída

input('VARIÁVEL LINGUÍSTICA DA FUNÇÃO DE SAÍDA - DE ENTER \ n')

VLS = 1

input('NÚMERO DE TERMOS LINGUÍSTICOS DA VARIÁVEL DE SAÍDA - DE ENTER
\ n')

NTLS = input('Entre com o número de Termos Linguísticos da Saída, NTLS = ')

% PARÂMETROS DOS TERMOS LINGUÍSTICOS DA VARIÁVEL DE SAÍDA

caso3 = input ('Voce pode entrar diretamente com o arquivo contendo os parâmetros dos termos
linguísticos de saída. Neste caso, Digite 1')

if (caso3 == 1)

    input('Coloque o arquivo na Rotina PARAMETROS_TERMOS_LING_SAIDA - De
    ENTER')

end

PTLS = PARAMETROS_TERMOS_LING_SAIDA(NTLS, caso3)

if (size(PTLS,1) ≠ NTLS)

```

```

        input('ERRO - A dimensão da matriz não confere com o número total de termos
        linguísticos')

break;
end
% DADOS: Vetor que armazena em cada posição  $j$  o valor, para estudo, atribuído a variável
linguística  $j$ 

input(Atribua um valor para cada variável de entrada dentre seus respectivos domínios - DE
ENTER \ n')
for i = 1:1:NVLE

    i
    DADOS(i) = input('Dado referente à variável linguística de entrada = ');

end
DADOS

% Gráfico das funções de pertinência
linf = 1;
lisup = 0;
flag = 1;
for i = 1:1:NVLE

    lisup = NTLE(i)+ lisup;
    g = gráficos(linf,lisup,PTLE,flag);
    linf = lisup + 1;
    flag = flag+1;

end
% Gráfico das funções de pertinência da variável de saída.
h = graf_variavel_saida(NTLS,PTLS);
% Base de Regras
% Regras: matriz que irá armazenar todas as combinações possíveis entre os termos linguísticos
de cada variável linguística de entrada.
Blocos = 1;
for i= 1:1:NVLE-1

```

```

Blocos = Blocos*NTLE(i);
end
for l= 1:1:NTLE(NVLE)

    V_AUX(l) = l;

end
V_AUX
% Construção da última coluna da matriz Regras
L_INF = 0;
for j = 1:1:Blocos

    for k = 1:1:NTLE(NVLE)
        Linha = L_INF+k;
        Regras(Linha, NVLE) = V_AUX(k);
    end
    L_INF = L_INF+NTLE(NVLE);

end
% Construção das demais colunas da matriz Regras.
for k= NVLE-1:-1:1

    L_INF = 0;
    for j= 1:1:Blocos/NTLE(k)

        for l= 1:1:NTLE(k)

            for m= 1:1: NTLE(NVLE)
                Linha = L_INF + m;
                Regras(Linha,k) = l;
            end
            L_INF = L_INF+NTLE(NVLE);

        end

    end

end

end
% Regras

```

```

caso4 = input ('Voce pode entrar diretamente com o arquivo contendo a SAÍDA DA BASE DE
REGRAS. Neste caso, Digite 1')
if (caso4 == 1)
input('Coloque o arquivo na Rotina SAIDA_BASE_DE_REGRAS - De ENTER')
end
SBR = SAIDA_BASE_DE_REGRAS(Regras,NVLE,NTLE,caso4)
dados_saida = PROC_DADOS_M(DADOS, NTLE, NVLE, NTLS, SBR, Regras, PTLE)
x_ini = PTLS(1,1);
x_fin = PTLS(NTLS,4);
DFZ = DEFUZZY_M(x_ini,x_fin,N_SUBINTERV,NTLS,PTLS,dados_saida)

```

1. Rotina de entrada de dados - Termos linguísticos de Entrada

% Caso você escolheu colocar o arquivo de dados diretamente aqui nesta rotina, vá para a linha ***

```
function numero_termos_ling = TERMOS_LING_ENTRADA(NVLE, caso)
```

% Esta função depende somente do número de variáveis linguísticas de entrada.

% Para cada variável linguística, o usuário deverá informar o número de termos linguísticos associados.

```
if (caso ==1)
```

% Lembre-se de que o número de coordenadas deve ser igual ao número de variáveis linguísticas.

```
    numero_termos_ling = [- - ... - ]; % LINHA *** - COLOQUE O SEU AR-
    QUIVO AQUI!
```

```
else
```

```
input('entre com o Número de Termos Linguísticos - DE ENTER')
```

```
input('associado a variável linguística - APERTE A TECLA ENTER')
```

```
for i = 1:1:NVLE
```

```
    i
```

```
    NTLE(i) = input('Número de Termos Linguísticos = ');
```

```

end
numero_termos_ling = NTLE;
end

```

2. Rotina de entrada de dados - Parâmetros dos Termos linguísticos de Entrada

% Caso você escolheu colocar o arquivo de dados diretamente aqui nesta rotina, vá para a linha***

```

function parametros_termos_ling = PARAMETROS_TERMOS_LING_ENTRADA(NVLE,
NTLE, caso2)

```

% Esta função depende do número de variáveis linguísticas de entrada e do número de termos linguísticos de entrada.

% PTLE: matriz que irá armazenar os parâmetros que darão origem as funções de pertinência trapezoidal ou triangular.

% Para se construir as funções de pertinência são necessários 4 parâmetros, que serão armazenados em uma matriz PTLE.

% Para cada termo linguístico, o usuário deverá informar os 4 parâmetros que darão origem à função de pertinência associada a esse termo linguístico.

% O usuário deverá digitar em cada linha da matriz PTLE os 4 parâmetros de cada termo linguístico separados por espaçamento simples e as linhas da matriz PTLE separadas por ponto e vírgula.

```

if (caso2 == 1)

```

```

    parametros_termos_ling = [- - - ; - - - ; ... ; - - - ]; %LINHA *** -
    COLOQUE O SEU ARQUIVO AQUI!

```

```

else

```

```

L_INF = 1;

```

```

L_SUP = 0;

```

```

for i=1:1:NVLE

```

```

    input('Início da entrada dos parâmetros associados à Variável - DE ENTER')

```

```

    i

```

```

    L_SUP = NTLE(i)+L_SUP;

```

```

for k = L_INF:1:L_SUP

    input('PARA CADA GRUPO DE TERMOS LINGUÍSTICOS
    - DE ENTER')
    k
    input('ENTRE COM OS 4 PARÂMETROS PARA A FUNÇÃO
    DE PERTINÊNCIA - DE ENTER')
    PTLE_AUX = input('4 VALORES ENTRE COLCHETES SE-
    PARADOS POR ESPAÇAMENTO SIMPLES = ');
    for j= 1:1:4
        PTLE(k,j) = PTLE_AUX(j);
    end
    L_INF =L_SUP + 1;
end

end

parametros_termos_ling = PTLE;

end

```

3. Gráficos das funções de pertinência das variáveis linguísticas de entrada

```

function g = gráficos(linf, lisup,PTLE,flag)

% Os pontos do eixo y serão sempre do tipo Y = [0 1 1 0].
tt=flag; % parâmetro utilizado para imprimir o título no gráfico.
Y = [0 1 1 0];
L_INF = linf;
L_SUP = lisup;
figure;
for j = L_INF:1:L_SUP

    for k = 1:1:4
        XX(k) = PTLE(j,k);
    end
    if(tt == 1)

```

```

plot(XX,Y);
title('Título para a Variável 1');
xlabel(' abscissa x');
ylabel('  $f_p(x)$  ');
axis([- 0 1.1]); limite dos eixos.
legend('Termo_ling(1)', 'Termo_ling(2)',..., 'Termo_ling(n)');
end
if(tt == 2)
plot(XX,Y);
title('Título para a Variável 2');
xlabel(' abscissa x ');
ylabel('  $f_p(x)$  ');
axis([- 0 1.1]); limite dos eixos
legend('Termo_ling(1)', 'Termo_ling(2)',..., 'Termo_ling(n)');
:
if(tt == j)
plot(XX,Y);
title('Título para a Variável j');
xlabel(' abscissa x');
ylabel('  $f_p(x)$  ');
axis([- 0 1.1]); % limite dos eixos
legend('Termo_ling(1)', 'Termo_ling(2)',..., 'Termo_ling(n)');
end
hold on;

end

```

4. Rotina de entrada de dados - Parâmetros dos Termos linguísticos de Saída

% Caso você escolheu colocar o arquivo de dados diretamente aqui nesta rotina, vá para a linha***

```

function parametros_termos_ling = PARAMETROS_TERMOS_LING_SAIDA(NTLE,
caso4)

```

% Esta função depende do número de termos linguísticos de saída.

% PTLS: matriz que irá armazenar os parâmetros que darão origem às funções de pertinência trapezoidal ou triangular.

% Para se construir as funções de pertinência são necessários 4 parâmetros, que serão armazenados em uma matriz PTLS.

% Para cada termo linguístico, o usuário deverá informar os 4 parâmetros que darão origem à função de pertinência associada a esse termo linguístico. O usuário deverá digitar em cada linha da matriz PTLS os 4 parâmetros de cada termo linguístico separados por espaçamento simples e as linhas da matriz PTLS separadas por ponto e vírgula.

if (caso4 == 1)

parametros_termos_ling = [- - - - ; - - - - ; ... ; - - - -]; LINHA *** -
COLOQUE O SEU ARQUIVO AQUI!

else

L_INF = 1;

input('Início da entrada dos parâmetros associados à Variável de Saída - DE ENTER')

for k = L_INF:1:NTLS

input('PARA CADA UM DOS TERMOS LINGUÍSTICOS - DE ENTER')

k

input('ENTRE COM OS 4 PARÂMETROS PARA A FUNÇÃO DE PERTINÊNCIA - DE ENTER')

PTLS_AUX = input('4 VALORES ENTRE COLCHETES SEPARADOS POR ESPAÇAMENTO SIMPLES = ');

for j = 1:1:4

PTLS(k,j) = PTLS_AUX(j); end

end

parametros_termos_ling = PTLS;

end

5. Gráfico das funções de pertinência da variável linguística de saída

function h = graf_variavel_saida(NTLS,PTLS)

% Os pontos do eixo y serão sempre do tipo $Y = [0 \ 1 \ 1 \ 0]$.

$Y = [0 \ 1 \ 1 \ 0];$

$L_INF = 1;$

$L_SUP = NTLS;$

figure;

for $j = L_INF:1:L_SUP$

for $k = 1:1:4$

$XX(k) = PTLS(j,k);$

end

plot(XX,Y);

axis($[- \ 0 \ 1.1]$); limite dos eixos.

title('Título para a Variável Saída');

hold on;

end

$h = 1;$

legend('Termo_ling(1)', 'Termo_ling(2)', ..., 'Termo_ling(n)');

6. Rotina para a determinação das Funções de pertinência trapezoidal ou triangular

function $f_p = \text{FUNC_PERT}(VV,DE)$

% VV é o vetor com os parâmetros da função de pertinência trapezoidal ou triangular - possui 4 parâmetros.

% DE é o dado onde a função será avaliada;

% Portanto esta função vai retornar o valor da função de pertinência em DE , ou seja, $f_p(DE)$.

if $(VV(1) < DE \ \& \ DE < VV(2))$

$f_p = (DE - VV(1))/(VV(2)-VV(1));$

elseif $(VV(2) \leq DE \ \& \ DE \leq VV(3))$

$f_p = 1;$

```

elseif (VV(3) < DE & DE < VV(4))

     $f_p = (VV(4) - DE)/(VV(4) - VV(3));$ 

else

     $f_p = 0;$ 

end

```

7. Rotina da saída da matriz base de regras - SAÍDA BASE DE REGRAS

% Caso você escolheu colocar o arquivo de dados diretamente aqui nesta rotina, vá para a linha***

```

function SBR = SAIDA_BASE_DE_REGRAS(Regras,NVLE,NTLE,caso3)

```

% Os valores associados aos termos linguísticos estão numerados com números naturais ordenados de forma crescente. Esta função depende da matriz Regras onde estão armazenados todas as combinações possíveis entre os termos linguísticos de cada variável de entrada. Depende do numero de variáveis linguísticas de entrada e do número de termos linguísticos de entrada.

```

if (caso3 == 1)

```

```

    SBR = [- ; - ; ... ; -]; %LINHA **** - COLOQUE O SEU ARQUIVO AQUI!

```

```

else

```

```

    Blocos = 1;

```

```

    for i= 1:1:NVLE-1

```

```

        Blocos= Blocos*NTLE(i);

```

```

    end

```

```

    Blocos = Blocos*NTLE(NVLE); %numero de linhas da matriz Regras

```

```

    for kk =1:1:Blocos

```

```

        input('Para cada regra de entrada - DE ENTER')

```

```

        for jj = 1:1:NVLE

```

```

            AUX_R(jj) = Regras(kk,jj);

```

```

end AUX_R
SBR(kk) = input('De um valor de saída variando de 1 ate NTLS, SBR = ');

end
SBR;
end

```

8. Rotina referente ao Processamento de dados

% Primeira etapa - Obtenção dos valores mínimos, de acordo com a base de regras, associados às variáveis linguísticas.

```
function valor_max_min = PROC_DADOS_M(DADOS, NTLE, NVLE, NTLS, Saida_Regras,
Regras, Vetor_Parametros)
```

% O vetor DADOS contém os valores associados a cada uma das variáveis linguísticas de entrada. Por exemplo, se o problema tiver 4 variáveis linguísticas de entrada, então o usuário devera informar 4 valores.

% O objetivo final será obter um valor de saída (defuzzificação) correspondente aos dados de entrada.

% O vetor NTLE contém a quantidade de termos linguísticos correspondente a cada uma das variáveis linguísticas.

% NVLE é o número de variáveis linguísticas da entrada.

% NTLS é o número de termos linguísticos de saída.

% O vetor Saida_Regras contém os termos linguísticos associados a cada uma das bases de regra.

% A matriz Regras contém todas as bases de regra.

% O vetor de parâmetros - Vetor_Parametros - contém os parâmetros que darão origem à função de pertinência.

% Esta função vai ser avaliada em um dos pontos do vetor DADOS.

% A variável Total_Regras contém o número total de regras que foram criadas. Este número dependera do número de variáveis linguísticas e do número de termos linguísticos associados a cada variável linguística.

```
Total_Regras = 1;
```

```
for ii=1:1:NVLE
```

```
Total_Regras = Total_Regras*NTLE(ii);
```

```
end
```

```
for nn=1:1:NTLS
```

```

VMax(nn) = 0; VMaxMin_S(nn)=0;

end
for i = 1:1:Total_Regras

    VMi=2;
    IND_saida = Saida_Regras(i);
    for j = 1:1:NVLE

        TL = Regras(i, j);
        DE = DADOS(j);
        VL = j;
        Linha = 0;
        for k = 1:1:j-1
            Linha = Linha + NTLE(k);
        end
        Linha = Linha + TL;
        for kk = 1:1:4
            VP_aux(kk) = Vetor_Parametros(Linha, kk); Vetor de Parâmetros aux-
            iliar - armazena quatro valores associados a um termo linguístico.
        end
        V_AUX = FUNC_PERT(VP_aux, DE);
        if (V_AUX < VMi)
            VMi = V_AUX;
        end

    end

    VMaxMin(IND_saida) = VMi;
    if (VMaxMin(IND_saida) > VMax(IND_saida))
        VMaxMin_S(IND_saida) = VMaxMin(IND_saida);
        VMax(IND_saida) = VMaxMin(IND_saida);
    end

end

end

valor_max_min = VMaxMin_S;

```

9. Rotina que irá calcular o valor da função de pertinência da região delimitada por uma poligonal

% A região é obtida através da união de conjuntos que surgem no processo de defuzzificação. A etapa final da defuzzificação vai gerar um número que representara o centro de gravidade da região anterior. Esta conta será feita na Rotina “PROC_DADOS_M” definida acima.

```
function GP = PERT_DEFUZ(NTLS, PTLS, VMAX_MIN, x)
```

% NTLS é o número de termos linguísticos da variável de saída. O vetor PTLS contém os parâmetros associados a cada um dos termos linguísticos da variável ling. de saída. O vetor VMAX_MIN contém os graus de pertinência limites de cada termo linguístico da variável de saída. Cada um dos graus de pertinência foi obtido a partir de um processo “maximo_minimo” que gera um valor constante (valor de corte) que representara o grau máximo de pertinência associado a um dado termo linguístico. Isto foi implementado na rotina PROC_DADOS. A variável x representa um ponto, no eixo horizontal, onde será calculada a função de pertinência da região limitada por uma poligonal (descrita acima).

```
GP_MAX = 0;
for ii=1:1:NTLS

    for jj=1:1:4
        VV(jj) = PTLS(ii,jj);
    end

    if (PTLS(ii,1) < x & x < PTLS(ii,4))

        GP_AUX = FUNC_PERT(VV,x);
        if (GP_AUX < VMAX_MIN(ii))
            GP_MIN = GP_AUX;
        else
            GP_MIN=VMAX_MIN(ii);
        end
    else
        GP_MIN=0;
    end

end

if (GP_MIN > GP_MAX)
```

```
GP_MAX = GP_MIN;
```

```
end
```

```
end
```

```
GP = GP_MAX;
```

10. Rotina DEFUZZY_M

% Esta Rotina vai calcular o centro de gravidade da região poligonal descrita na Rotina PERT_DEFUZ.

% O centro de gravidade será calculado pela formula: $\text{integ}\{xf_p(x)dx\}/\text{integ}\{f_p(x)dx\}$. As integrais serão calculadas pela Regra de Simpson. $\mu(x)$ é a função de pertinência associada a região delimitada por uma poligonal. O valor desta função é calculada pela Rotina PERT_DEFUZ.

function DFZ = DEFUZZY_M(x_ini, x_fin, N_SUBINT, NTLS, PTLS, dados_saida)

x_ini % é o extremo inferior do intervalo de integração.

x_fin % é o extremo superior do intervalo de integração.

N_SUBINT % é o número de subintervalos utilizados na regra de integração.

% NTLS é o número de termos linguisticos da variável de saída.

% A matriz PTLS contém os parâmetros de todos os termos linguísticos de saída.

% O vetor dados_saida contém os graus de pertinência limites de cada termo linguístico da variável de saída.

```
n= N_SUBINT;
```

```
h = (x_fin - x_ini)/n; % n = número de subintervalos de [x_ini , x_fin]
```

```
SD1= PERT_DEFUZ(NTLS, PTLS, dados_saida, x_ini); Soma no denominador
```

```
U_p(1)=SD1; % vetor que vai armazenar a função de pertinência associada ao processo de defuzzificação
```

```
SN1= x_ini*SD1; % Soma no numerador
```

```
SD2 = PERT_DEFUZ(NTLS, PTLS, dados_saida, x_fin);
```

```
U_p(n+1)=SD2;
```

```
SN2 = x_fin*SD2;
```

```
SN_I=0;
```

```
SD_I=0;
```

```
X(1) = x_ini; % vetor utilizado para fazer o gráfico da poligonal
```

```

X(n+1) = x_fin;
figure;
for j = 1:2:n-1

    x_j= x_ini + j*h;
    X(j+1)= x_j;
    Pert_aux = PERT_DEFUZ(NTLS, PTLS, dados_saida, x_j);
    U_p(j+1) = Pert_aux;
    SN_I = SN_I + x_j*Pert_aux;
    SD_I = SD_I + Pert_aux;

end
SN_I;
SD_I;
SN_P=0;
SD_P=0;
for j = 2:2:n-2

    x_j= x_ini + j*h;
    X(j+1)= x_j;
    Pert_aux = PERT_DEFUZ(NTLS, PTLS, dados_saida, x_j);
    U_p(j+1) = Pert_aux;
    SN_P = SN_P + x_j*Pert_aux;
    SD_P = SD_P + Pert_aux;

end
SN_P;
SD_P;
CG = (SN1 + 4*SN_I + 2*SN_P + SN2)/(SD1 +4*SD_I + 2*SD_P + SD2);
plot(X,U_p)
area(X,U_p);
title('F. de pertinencia:defuzzificacao');
axis([x_ini x_fin 0 1]);
DFZ = CG;

```

Apêndice 2

Para o método de Takagi-Sugeno utilizando funções de pertinência do tipo gaussiano aproveitaremos algumas rotinas criadas para o método de Mamdani, com pequenas mudanças. Não citaremos as rotinas **1, 2 e 7** do Apêndice 1, pois estas continuam iguais para o método de Takagi-Sugeno.

Código Principal para o Método de Takagi-Sugeno; Utilizando Funções de Pertinência Gaussiana

```
% Entrada de dados
% NVLE : Número de variaveis linguisticas das funcoes de entrada
clear;
N_SUBINTERV = 10000; % Número de subintervalos para a defuzzificacao.
input('VARIAVEIS LINGUISTICAS DAS FUNCOES DE ENTRADA - DE ENTER  n')
NVLE = input('Entre com o numero de variaveis linguisticas: ')
caso = input('Voce pode entrar diretamente com o arquivo contendo os termos linguisticos.
Neste caso, DIGITE 1')

if (caso == 1)

    input('Coloque o arquivo na Rotina TERMOS_LING_ENTRADA - De ENTER')

end

NTLE = TERMOS_LING_ENTRADA(NVLE,caso) Número de termos linguísticos de entrada.
Length_NTLE = length(NTLE);

if (Length_NTLE == NVLE)
```

```

        input('ERRO - A dimensao do vetor nao confere com o numero de variaveis linguis-
        ticas')

break;
end
NL_PTLE = 0; % Número de linhas da matriz PTLE
for i = 1:1:Length_NTLE

    NL_PTLE = NL_PTLE + NTLE(i);

end
caso2 = input ('Voce pode entrar diretamente com o arquivo contendo os parametros dos termos
linguisticos de entrada. Neste caso, Digite 1')
if (caso2 == 1)

    input('Coloque o arquivo na Rotina PARAMETROS_TERMOS_LING_ENTRADA
    - De ENTER')

end
PTLE = PARAMETROS_TERMOS_LING_ENTRADA(NVLE, NTLE, caso2)
if (size(PTLE,1) == NL_PTLE)

    input('ERRO - A dimensao da matriz nao confere com o numero total de termos
    linguisticos')

break;
end

input('ATRUBUA UM VALOR PARA CADA VARIABEL DE ENTRADA DENTRE
OS SEUS RESPECTIVOS DOMINIO - DE ENTER  n')

for i = 1:1:NVLE

    i
    DADOS(i) = input('Dado referente aa variavel linguistica de entrada = ');

end
DADOS

```

```
% Gráfico das funcoes de pertinencia
```

```
linf = 1;
```

```
lisup = 0;
```

```
flag = 1;
```

```
for i = 1:1:NVLE
```

```
    lisup = NTLE(i)+lisup;
```

```
    g = graficos(linf,lisup,PTLE,flag);
```

```
    linf = lisup + 1;
```

```
    flag = flag+1;
```

```
end
```

```
% Base de Regras
```

```
% Regras: matriz que irá armazenar todas as combinações possíveis entre os termos linguísticos  
de cada variável linguística de entrada.
```

```
Blocos = 1;
```

```
for i= 1:1:NVLE-1
```

```
    Blocos= Blocos*NTLE(i);
```

```
end
```

```
for l= 1:1:NTLE(NVLE)
```

```
    V_AUX(l)= 1;
```

```
end
```

```
% Construção da última coluna da matriz Regras
```

```
L_INF = 0;
```

```
for j= 1:1:Blocos
```

```
    for k= 1:1:NTLE(NVLE)
```

```
        Linha= L_INF+k;
```

```
        Regras(Linha, NVLE)= V_AUX(k);
```

```
    end
```

```
    L_INF= L_INF+NTLE(NVLE);
```

```

end
% Contrução das demais colunas da matriz Regras.
for k= NVLE-1:-1:1

    L_INF= 0;
    for j= 1:1:Blocos/NTLE(k)

        for l= 1:1:NTLE(k)

            for m= 1:1: NTLE(NVLE)
                Linha= L_INF + m;
                Regras(Linha,k)= l;
            end
            L_INF= L_INF+NTLE(NVLE);
        end
    end
end

end

% Entrada de dados - SUGENO
input('NUMERO DE FUNCOES DE SUGENO PARA A SAIDA - DE ENTER  n')
NFS = input('Entre com o numero de FUNCOES de SUGENO para a saida, NFS = ') Número
de funções de Sugeno.
caso3 = input ('Voce pode entrar diretamente com o arquivo contendo a SAIDA DA BASE DE
REGRAS. Neste caso, Digite 1')
if (caso3 == 1)

    input('Coloque o arquivo na Rotina SAIDA_BASE_DE_REGRAS - De ENTER')

end

SBR = SAIDA_BASE_DE_REGRAS(Regras,NVLE,NTLE,caso3);
Regras = [Regras SBR] % Matriz que contém todas as regras.
valor_min = PROC_DADOS_S(DADOS, NTLE, NVLE, NFS, SBR, Regras, PTLE).
defuzz_Sugeno = DEFUZZ_S(DADOS, NTLE, NVLE, NFS, SBR, valor_min).

```

1. Rotina para a determinação das Funções de pertinência gaussiana

```
function fp = FUNC_PERT(VV,DE)
```

% Função gaussiana do tipo $H(x) = \exp(\alpha/(x^2 - 1))$, x em $[-1, 1]$; $\alpha > 0$.

```

% Será utilizada a seguinte transformação  $T(x) = c_a.x + ti$  de  $[a, b]$  em  $[-1, 1]$ 
% A função gaussiana utilizada será  $G(x) = H(T(x))$ .
% VV é o vetor com os parâmetros da função de pertinência gaussiana - possui 3
parâmetros onde o primeiro parâmetro equivale ao início do intervalo da função, o se-
gundo parametro equivale ao final do intervalo da função,o terceiro parametro modifica
a forma da gaussiana deixando-a mais fina ou mais larga.
% DE é a variável de entrada; a função de pertinência será avaliada neste ponto. Portanto
esta função vai retornar o valor da função de pertinência em DE, ou seja,  $fp(DE)$ .

a = VV(1);
b = VV(2);
alpha = VV(3); % modifica a forma da gaussiana deixando-a mais fina ou mais larga.

c_a = 2.0/(b-a); % coeficiente angular da transformação:  $[a, b] \rightarrow [-1, 1]$ 
ti = -(b+a)/(b-a); % termo independente da transformação acima.

eps =0.00001;
A = a+eps;
B = b-eps;

x_1 = (a+b)/2;
x_T = (c_a*x_1 + ti);
x_T = x_T*x_T -1;
cc_a = exp(-alpha/x_T);
if (A ≤ DE & DE ≤ B)

    xx_T = c_a*DE + ti;
    xx_T = xx_T.*xx_T;
    fp = cc_a*exp(alpha./(xx_T -1 ));

else

    fp = 0;

```

end

2. Rotina para determinar as funções de saídas (Sugeno)

```
% Funcao de Sugeno
function f_sug = SUGENO(nn, DADOS)
% A variável nn corresponde ao número da função de Sugeno que esta sendo considerada
para a variável de saída.
if nn == 1

    f_sug = determine aqui sua função de saída número 1;

elseif nn ==2

    f_sug = determine aqui sua função de saída número 2;

elseif nn ==3

    f_sug = determine aqui sua função de saída número 3;

:
elseif nn == p

    f_sug = determine aqui sua função de saída número ;

end
```

3. Gráficos das funções de pertinência das variáveis linguísticas de entrada

```
function g = graficos(linf, lisup,PTLE,flag)
% Os pontos do eixo y serao sempre do tipo Y = [0 1 1 0]
tt=flag; % parâmetro utilizado para imprimir o título no gráfico
Y = [0 1 1 0];
L_INF = linf;
L_SUP = lisup;
figure;
for j = L_INF:1:L_SUP

    for k = 1:1:3

        VV(k) = PTLE(j,k);
```

```

end
a = VV(1);
b = VV(2);
alpha = VV(3); % modifica a forma da gaussiana deixando-a mais fina ou mais
larga.
c_a = 2.0/(b-a); % coeficiente angular da transformacao: [a, b] → [-1 1]
ti = -(b+a)/(b-a); % termo independente da transformacao acima.
eps = 0.00001;
A = a+eps;
B = b-eps;
x_1 = (a+b)/2;
x_T = (c_a*x_1 + ti);
x_T = x_T*x_T - 1;
cc_a = exp(-alpha/x_T);
x = A:0.001:B;
xx_T = c_a*x + ti;
xx_T = xx_T.*xx_T;
y = cc_a*exp(alpha./(xx_T - 1 ));
if(tt == 1)
plot(x, y)
title('Título para a Variável 1');
xlabel('abscissa x');
ylabel('f_p(x)');
axis([- 0 1.1]); limite dos eixos
end
if(tt == 2)
plot(x, y)
title('Título para a Variável 2');
xlabel('abscissa x');
ylabel('f_p(x)');
axis([- 0 1.1]); limite dos eixos
end
:
if(tt == j)

```

```

    plot(x, y)
    title('Título para a Variável j');
    xlabel('abscissa x');
    ylabel('f_p(x)');
    axis([- 0 1.1]); limite dos eixos
end
hold on

end

g = 1;

```

4. Rotina referente ao Processamento de dados

```

% Processamento de dados.

% Primeira etapa - Obtenção dos valores mínimos, de acordo com a base de regras, asso-
ciados aas variáveis linguísticas.

function valor_min = PROC_DADOS_S(DADOS, NTLE, NVLE, NFS, Saida_Regras,
Regras, Vetor_Parametros)

% O vetor Dados contém os valores associados a cada uma das variáveis linguísticas de
entrada. Por exemplo, se o problema tiver 4 variáveis linguísticas de entrada, então o
usuário deverá informar 4 valores.

% O objetivo final será obter um valor de saída (defuzzificação) correspondente aos dados
de entrada.

% O vetor NTLE contém a quantidade de termos linguísticos correspondente a cada uma
das variáveis linguísticas.

% NVLE é o número de variáveis linguísticas da entrada.

% NFS é o número de funções de Sugeno utilizadas para representar a função de saída.

% O vetor Saida_Regras contém o índice da função de Sugeno associada a cada uma das
bases de regra.

% A matriz Regras contém todas as bases de regra.

% O vetor de parâmetros - Vetor_Parametros - contém os parâmetros que darão origem
à função de pertinência.

% Esta função vai ser avaliada em um dos pontos do vetor DADOS.

% A variável Total_Regras contém o número total de regras que foram criadas. Este
número dependerá do número de variáveis linguísticas e do número de termos linguísticos

```

associados a cada variável linguística.

```
Total_Regras = 1;
```

```
for ii=1:1:NVLE
```

```
Total_Regras = Total_Regras*NTLE(ii);
```

```
end
```

```
for nn=1:1:NFS
```

```
    VSug(nn) = SUGENO(nn, DADOS); % valor da função de Sugeno de indice
```

```
    nn (nn variando de 1 ate NFS)
```

```
end
```

```
for i = 1:1:Total_Regras
```

```
    VMin(i)=2;
```

```
    IND_saida = Saida_Regras(i);
```

```
    for j = 1:1:NVLE
```

```
        TL = Regras(i, j);
```

```
        DE = DADOS(j);
```

```
        VL = j;
```

```
        Linha = 0;
```

```
        for k = 1:1:j-1
```

```
            Linha = Linha + NTLE(k);
```

```
        end
```

```
        Linha = Linha + TL;
```

```
        for kk = 1:1:3
```

```
            VP_aux(kk) = Vetor_Parametros(Linha, kk); % Vetor de Parâmetros
```

```
            auxiliar - armazena quatro valores associados a um termo ling.
```

```
        end
```

```
        V_AUX = FUNC_PERT(VP_aux, DE);
```

```
        if (V_AUX < VMin(i))
```

```
            VMin(i) = V_AUX; end
```

```
    end
```

```
end
```

```
valor_min = VMin;
```

5. Rotina referente à Defuzzificação

% Cálculo da média ponderada - método de Sugeno

function defuzz_Sugeno = DEFUZZ_S(DADOS, NTLE, NVLE, NFS, Saida_Regras, valor_min)

Total_Regras = 1;

for ii=1:1:NVLE

Total_Regras = Total_Regras*NTLE(ii);

end

for nn=1:1:NFS

VSug(nn) = SUGENO(nn, DADOS); % valor da funcao de Sugeno de indice nn (nn variando de 1 ate NFS)

end

Soma_N = 0; % Soma no numerador

Soma_D = 0; % Soma no denominador

for i = 1:1:Total_Regras

IND_saida = Saida_Regras(i);

Soma_N = Soma_N + VMin(i)*VSug(IND_saida);

Soma_D = Soma_D + VMin(i);

end

defuzz_Sugeno = Soma_N/Soma_D;

Apêndice 3

Código para o Modelo 1: Decaimento de Fármaco

```
% Entrada de dados para o problema (Decaimento de Fármaco)
clear;
NVLE = 1 % o problema proposto possui somente uma variavel de entrada
NTLE = 2 % o número de termos linguístico para a variável de entrada, onde cada termo
linguístico será denotado por:
NTLE_1= 'TEMPO BAIXO';
NTLE_2= 'TEMPO ALTO';
Length_NTLE = length(NTLE);

    if (Length_NTLE  $\neq$  NVLE)
input('ERRO - A dimensão do vetor não confere com o número de variáveis linguísticas')
break;
end
NL_PTLE = 0; % Número de linhas da matriz PTLE.
for i = 1:1:Length_NTLE

    NL_PTLE = NL_PTLE + NTLE(i);

end
PTLE = [0 0 0 28.6;0 28.6 100 100] % matriz que contém os parâmetros que darão origem as
funções de pertinência associado a cada termo linguístico de entrada.
if (size(PTLE,1)  $\neq$  NL_PTLE)

    input('ERRO - A dimensão da matriz não confere com o número total de termos
linguísticos')

break;
end
```

```

% gráfico das funções de pertinência
linf = 1;
lisup = 0;
flag = 1;
for i = 1:1:NVLE

    lisup = NTLE(i)+lisup; g = graficos(linf,lisup,PTLE,flag); linf = lisup + 1; flag =
    flag+1;

end
% Base de Regras
% Regras: matriz que irá armazenar todas as combinações possíveis entre os termos linguísticos
de cada variável linguística de entrada
Blocos = 1;
for i= 1:1:NVLE-1

    Blocos= Blocos*NTLE(i);

end
for l= 1:1:NTLE(NVLE)

    V_AUX(l)= 1;

end
% Construção da última coluna da matriz Regras
L_INF = 0;
for j= 1:1:Blocos

    for k= 1:1:NTLE(NVLE)

        Linha= L_INF+k;
        Regras(Linha, NVLE)= V_AUX(k);

    end

    L_INF= L_INF+NTLE(NVLE);

end
% Contrução das demais colunas da matriz Regras
for k= NVLE-1:-1:1

```

```

L_INF= 0;
for j= 1:1:Blocos/NTLE(k)

    for l= 1:1:NTLE(k)

        for m= 1:1: NTLE(NVLE)
            Linha= L_INF + m;
            Regras(Linha,k)= l;
        end
        L_INF= L_INF+NTLE(NVLE);
    end

end

end

% Entrada de dados - SUGENO
NFS = 2 % NFS é o número de funções de Sugeno para a saída do sistema
SBR = [1;2] % SBR é o vetor saída da base de regras, onde o elemento na posicao  $x_{11}$  indica
a saída para a regra de número 1 e o elemento na posicao  $x_{12}$  indica a saída para a regra de
numero 2, o elemnto  $x_{11} = 1$  corresponde a função de sugeno de número 1 e o elemento  $x_{12} = 2$ 
coresponde a função de sugeno de número 2.
valo_min = PROC_DADOS_S(DADOS, NTLE, NVLE, NFS, SBR, Regras, PTLE);
% Representação gráfica da Solução Analítica e da Solução Fuzzy do problema proposto (De-
caimento de Fármaco).
n = 10000; % número de divisões para o intervalo da função de entrada.
x_ini = PTLE(1,1); início do intervalo
x_fin = PTLE(NTLE,4); fim do intervalo
h = (x_fin - x_ini)/n;
U_p(1) = DEFUZZ_S(x_ini, NTLE, NVLE, NFS, SBR, Regras, vlor_min);
U_p(n+1) = DEFUZZ_S(x_fin, NTLE, NVLE, NFS, SBR, Regras, vlor_min);
X(1) = x_ini; % vetor utilizado para fazer o gráfico da solucao fuzzy
X(n+1) = x_fin;
figure;
for j = 1:1:n-1

    x_j = x_ini + j*h;
    dados_aux = DEFUZZ_S(x_j, NTLE, NVLE, NFS, SBR, Regras, vlor_min);

```

```

X(j+1)= x_j;
U_p(j+1) = dados_aux;

end

y = 1.3*exp(-0.173*X) + 0.82*exp(-0.0092*X);
plot(X,y,X,U_p)
xlabel('horas');
ylabel('Concentração sanguínea(mg/l)');
axis([0 100 0.2 2.2]);
legend('C(t)solução clássica', 'C(t)solução fuzzy');

```

1. Rotina para determinar as funções de saídas (Sugeno)

Função de Sugeno

function f_sug = SUGENO(nn, DADOS)

if nn == 1

$$f_{sug} = 10^{(0.255-0.021*DADOS)};$$

elseif nn ==2

$$f_{sug} = 10^{(-0.086-0.004*DADOS)};$$

end

Apêndice 4

Código para o Modelo 2: Modelo *Fuzzy* para a Densidade do Solo

```
%Entrada de dados para o problema (Densidade do Solo)
%NVLE : Número de variáveis linguísticas das funções de entrada clear;
NVLE = 2
N.SUBINTERV = 10000; número de subintervalos para a defuzzificação
% caso = input('Você pode entrar diretamente com o arquivo contendo os termos linguísticos.
Neste caso, DIGITE 1')
% if (caso == 1)

    % input('Coloque o arquivo na Rotina TERMOS_LING_ENTRADA - De ENTER')

% end
% NTLE = TERMOS_LING_ENTRADA(NVLE,caso)
NTLE = [3 3]
Length_NTLE = length(NTLE);

if (Length_NTLE ≠ NVLE)

    input('ERRO - A dimensao do vetor não confere com o número de variáveis linguísticas')

break;
end
NL_PTLE = 0;
for i = 1:1:Length_NTLE

    NL_PTLE = NL_PTLE + NTLE(i);
```

```

end
% caso2 = input ('Você pode entrar diretamente com o arquivo contendo os parâmetros dos
termos linguísticos de entrada. Neste caso, Digite 1')
% if (caso2 == 1)

    % input('Coloque o arquivo na Rotina PARAMETROS_TERMOS_LING_ENTRADA
    - De ENTER')

% end
% PTLE = PARAMETROS_TERMOS_LING_ENTRADA(NVLE, NTLE, caso2)
PTLE = [-1109.02 7288.22 1399.54;-1680.62 6716.62 1399.54;1440.98 9838.22 1399.54; 18.50
34.00 2.435; 22.97 37.39 2.44;12.65 27.85 2.45]
if (size(PTLE,1)  $\neq$  NL_PTLE)

    input('ERRO - A dimensão da matriz não confere com o número total de termos
    linguísticos')

break;
end
input('ATRUBUA UM VALOR PARA CADA VARIÁVEL DE ENTRADA DENTRE OS
SEUS RESPECTIVOS DOMÍNIO - DE ENTER
n')
for i = 1:1:NVLE

    i
    DADOS(i) = input('Dado referente à variável linguística de entrada = ');

end
DADOS
gráfico das funções de pertinência
linf = 1;
lisup = 0;
flag = 1;
for i = 1:1:NVLE

    lisup = NTLE(i)+lisup;
    g = graficos(linf,lisup,PTLE,flag);
    linf = lisup + 1;
    flag = flag+1;

```

```

end
% Base de Regras
% Regras: matriz que irá armazenar todas as combinações possíveis entre os termos linguísticos
de cada variável linguística de entrada
% caso3 = input ('Você pode entrar diretamente com o arquivo contendo a BASE DE REGRAS.
Neste caso, Digite 1')
% if (caso3 == 1)

    % input('Coloque o arquivo na Rotina MATRIZ_DA_BASE_DE_REGRAS - De EN-
    TER')

% end
N_Regras = input('entre com o número total de regras N_Regras: ')
Regras = MATRIZ_DA_BASE_DE_REGRAS(NVLE,caso3)
Regras = [1 1;2 2;3 3];
% Entrada de dados - SUGENO
% input('NUMERO DE FUNÇÕES DE SUGENO PARA A SAIDA - DE ENTER
n')
% NFS = input('Entre com o número de FUNÇÕES de SUGENO para a saída, NFS = ')
NFS= 3
% caso4 = input ('Você pode entrar diretamente com o arquivo contendo a SAIDA DA BASE
DE REGRAS. Neste caso, Digite 1')
% if (caso4 == 1)

    % input('Coloque o arquivo na Rotina SAIDA_BASE_DE_REGRAS - De ENTER')

% end
% SBR = SAIDA_BASE_DE_REGRAS(Regras,NVLE,NTLE,caso4)
SBR = [1;2;3];
Regras = [Regras  SBR]
valo_min = PROC_DADOS_S(DADOS, N_Regras, NTLE, NVLE, NFS, SBR, Regras, PTLE);
defuzz_Sugeno = DEFUZZ_S(DADOS, NTLE, NVLE, NFS, SBR, valor_min)

```

1. Rotina para determinar a Matriz da Base de Regras

```

% Construção da matriz Regras.
% Esta matriz contém todas as regras envolvendo as variáveis linguísticas de entrada.
% CASO VOCÊ ESCOLHEU COLOCAR O ARQUIVO DE DADOS DIRETAMENTE

```

AQUI NESTA ROTINA, VA PARA A LINHA ***

```
function Regras = MATRIZ_DA_BASE_DE_REGRAS(NVLE,caso3)
if (caso3 == 1)

    Regras = [1 1;2 2;3 3]; %LINHA **** - COLOQUE O SEU ARQUIVO AQUI!

else
input('CONSTRUÇÃO DA REGRAS RELACIONADAS COM AS VARIÁVEIS LINGUÍSTICAS
- DE ENTER
n')
N_REGRAS = input('Entre com o número total de regras: ')
for i=1:1:N_REGRAS

    input('ÍNICIO DA ELABORAÇÃO DAS REGRAS - DE ENTER')
    i
    input('PARA CADA LINHA DA MATRIZ REGRAS - DE ENTER')
    input('ENTRE COM OS NÚMEROS ASSOCIADOS AOS TERMOS LINGUÍSTICOS
DAS VARIÁVEIS DE ENTRADA QUE IRÃO COMPOR CADA LINHA DA
MATRIZ REGRAS - DE ENTER')
    Regras_AUX = input('ENTRE COM A COMBINAÇÃO ENTRE OS TER-
MOS LINGUÍSTICOS DAS VARIÁVEIS DE ENTRADA ENTRE COLCHETES
SEPARADOS POR ESPACAMENTO SIMPLES = ');
    for j= 1:1:NVLE
        Regras(i,j) = Regras_AUX(j);
    end

end

Regras
end
```

Apêndice 5

Código para o Modelo 5: Conversão de Assintomático para Sintomático. Modelos *Fuzzy* com λ Dependendo do Nível de CD4+ e da Carga Viral

```
% Entrada de dados
%NVLE : Numero de variaveis linguisticas das funcoes de entrada
clear;
N_SUBINTERV = 100000; % Numero de subintervalos que sera utilizado para a defuzzificao
input('VARIAVEIS LINGUISTICAS DAS FUNCOES DE ENTRADA - DE ENTER
n')
% NVLE = input('Entre com o numero de variaveis linguisticas: ')
NVLE = 2
%caso = input('Voce pode entrar diretamente com o arquivo contendo os termos linguisticos.
Neste caso, DIGITE 1')
%if (caso == 1)

    % input('Coloque o arquivo na Rotina TERMOS_LING_ENTRADA - De ENTER')

%end
%NTLE : Vetor que contem em cada posicao j o numero de termos linguisticos para a variavel
linguistica de entrada j
%NTLE = TERMOS_LING_ENTRADA(NVLE,caso)
NTLE = [3 5]
Length_NTLE = length(NTLE);
if (Length_NTLE  $\neq$  NVLE)
```

```

        input('ERRO - A dimensao do vetor nao confere com o numero de variaveis linguis-
        ticas')

break;

end

NL_PTLE = 0;

for i = 1:1:Length_NTLE

    NL_PTLE = NL_PTLE + NTLE(i);

end

%PTLE : Matriz que ira armazenar, respectivamente, em cada linha os parametros para a
construcao das funcoes de pertinencia associadas aos termos linguisticos.

%caso2 = input ('Voce pode entrar diretamente com o arquivo contendo os parametros dos
termos linguisticos de entrada. Neste caso, Digite 1')

%if (caso2 == 1)

    % input('Coloque o arquivo na Rotina PARAMETROS_TERMOS_LING_ENTRADA
    - De ENTER')

%end

%PTLE = PARAMETROS_TERMOS_LING_ENTRADA(NVLE, NTLE, caso2)
PTLE = [-0.1 -0.05 0.02 0.15;0.05 0.15 0.45 0.5;0.45 0.55 1.45 1.5;-0.1 -0.05 0.01 0.1;0.05 0.1
0.15 0.2;0.15 0.2 0.28 0.35; 0.25 0.3 0.35 0.5;0.45 0.5 1.45 1.5]
if (size(PTLE,1)  $\neq$  NL_PTLE)

    input('ERRO - A dimensao da matriz nao confere com o numero total de termos
    linguisticos')

break;

end

%grafico das funcoes de pertinencia

linf = 1;

lisup = 0;

flag = 1;

for i = 1:1:NVLE

```

```

lisup = NTLE(i)+lisup;
g = graficos(linf,lisup,PTLE,flag);
linf = lisup + 1;
flag = flag+1;

end
%input('ATRUBUA UM VALOR PARA CADA VARIABEL DE ENTRADA DENTRE OS
SEUS RESPECTIVOS DOMINIO - DE ENTER
n')
%for i = 1:1:NVLE

    % i
    % DADOS(i) = input('Dado referente aa variavel linguistica de entrada = ');

%end
%DADOS
%Base de Regras
%Regras: matriz que ira armazenar todas as combinacoes possiveis entre os termos linguisticos
de cada variavel linguistica de entrada.
Blocos = 1;
for i= 1:1:NVLE-1

    Blocos= Blocos*NTLE(i);

end
for l= 1:1:NTLE(NVLE)

    V_AUX(l)= 1;

end
V_AUX;
%Construcao da ultima coluna da matriz Regras
L_INF = 0;
for j= 1:1:Blocos

    for k= 1:1:NTLE(NVLE)

        Linha= L_INF+k;
        Regras(Linha, NVLE)= V_AUX(k);
    end
end

```

```

end
L_INF= L_INF+NTLE(NVLE);

end
%Construcao das demais colunas da matriz Regras
for k= NVLE-1:-1:1

    L_INF= 0;
    for j= 1:1:Blocos/NTLE(k)

        for l= 1:1:NTLE(k)

            for m= 1:1: NTLE(NVLE)
                Linha= L_INF + m;
                Regras(Linha,k)= l;
            end
            L_INF= L_INF+NTLE(NVLE);
        end
    end

end

end

end
Regras;
%Entrada de dados - SUGENO
%input('NUMERO DE FUNCOES DE SUGENO PARA A SAIDA - DE ENTER
n')
%NFS = input('Entre com o numero de FUNCOES de SUGENO para a saida, NFS = ')
NFS = 4;
%if (caso3 == 1)

    % input('Coloque o arquivo na Rotina SAIDA_BASE_DE_REGRAS - De ENTER')

%end
%SBR = SAIDA_BASE_DE_REGRAS(Regras,NVLE,NTLE,caso3)
SBR = [4;3;3;2;1;4;4;3;2;2;4;4;3;3;2];
Regras = [Regras SBR]
%CODIGO ROSANA - FAZER SIMILAR

```

```
CargaViral=[0 0.1 0.1 2 4 5 7 8 10 20 25 30 33 40 45 50 58 60 65 70 80 93 95 100 ... 101
130 200 320 400 550 610 700 810 1040 1200 1320 1400 1600 2000 2100 2200 2300 2500 2800
3000 ... 3050 3100 4500 5600 6000 6700 7000 7500 8000 8400 8900 9000 10000 13000 15000];
```

```
a=0.1;
```

```
m=1;
```

```
CD4 =(0.3)./(a + m*CargaViral);
```

```
CD4=CD4/(0.3/a);
```

```
CargaViral2=CargaViral/15000;
```

```
for i=1:length(CargaViral2)
```

```
    DADOS(1) = CargaViral2(i);
```

```
    DADOS(2) = CD4(i);
```

```
    valo_min = PROC_DADOS_S(DADOS, NTLE, NVLE, NFS, SBR, Regras, PTLE);
```

```
    lambda2(i) = DEFUZZ_S(DADOS, NTLE, NVLE, NFS, SBR, valor_min);
```

```
end
```

```
lambda2=lambda2';
```

```
% Solução Fuzzy
```

```
t=0:1:length(CargaViral2)-1;
```

```
for i=1:length(CargaViral2)
```

```
    for j=1:length(CargaViral2)
```

```
        A1(i,j)=-lambda2(j)*t(i);
```

```
        B1(i,j)=exp(A1(i,j));
```

```
        C1(i,j)=1-exp(A1(i,j));
```

```
    end
```

```
end
```

```
figure;
```

```
surf(lambda2,t,B1);
```

```
title('Solucao Fuzzy x');
```

```

view(-37.5,5);
xlabel('lambda');
ylabel('tempo');
zlabel('x(t)');
figure;
surf(lambda2,t,C1);
title('Solução Fuzzy y ');
view(-37.5,10);
xlabel('lambda');
ylabel('tempo');
zlabel('y(t)');
for j=1:length(CargaViral2)

    for i=1:length(CargaViral2)

        M1(i,1) = CargaViral2(j);
        M1(i,2) = CD4(i);
        DADOS =[M1(i,1) M1(i,2)];
        valo_min = PROC_DADOS_S(DADOS, NTLE, NVLE, NFS, SBR, Re-
            gras, PTLE);
        zz1(i,j) = DEFUZZ_S(DADOS, NTLE, NVLE, NFS, SBR, valor_min);

    end

end figure;
surf(CargaViral2,CD4,zz1);
rotate3d;
xlabel('Carga viral ( $v$ )');
ylabel('Nível de CD4+ ( $c$ )');
zlabel('Taxa de transferencia ( $\lambda$ )');

figure;
plot(CD4,zz1(:,2));
xlabel('Nível de CD4+ ( $c$ )');
ylabel('Taxa de transferencia ( $\lambda$ )');
figure;

```

```

plot(CD4,zz1(:,30));
xlabel('CD4+');
ylabel('lambda');
title('Corte na superficie com plano paralelo a CD4+ - Carga Viral(30)');
figure;
plot(CD4,zz1(:,length(CargaViral2)));
xlabel('CD4+');
ylabel('lambda'); title('Corte na superficie com plano paralelo a CD4+ - Carga Viral(60)');
figure;
plot(CargaViral2,zz1(1,:));
xlabel('Carga viral ( $v$ )');
ylabel('Taxa de transferencia ( $\lambda$ )');
figure;
plot(CargaViral2,zz1(40,:));
xlabel('Carga Viral');
ylabel('lambda');
title('Corte na superfceie com plano paralelo a Carga Viral - CD4(40)');
figure;
plot(CargaViral2,zz1(length(CargaViral2),:));
xlabel('Carga Viral');
ylabel('lambda');
title('Corte na superfceie com plano paralelo a Carga Viral - CD4(60)');
%FIM CODIGO ROSANA

```