



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-10296-TDI/915

**USO DE RELAÇÕES DE SIMILARIDADE PARA
TRATAMENTO DE CONSISTÊNCIA E COBERTURA EM
SISTEMAS DE REGRAS DIFUSAS**

Isabela Neves Drummond

Dissertação de Mestrado do Curso da Pós-Graduação em Computação Aplicada,
orientada pela Dra. Sandra Aparecida Sandri, aprovada em 26 de fevereiro de 2003.

681.3.019

DRUMMOND, I. N.

Uso de relações de similaridade para tratamento de consistência e cobertura em sistema de regras difusas / I. N. Drummond. – São José dos Campos: INPE, 2003.

79p. – (INPE-10296-TDI/915).

1.Sistema difuso. 2.Conjunto difuso. 3.Relação de similaridade. 4.Sistema implicativo. 5.Algoritmo genético. 6.Inteligência artificial. I.Título.

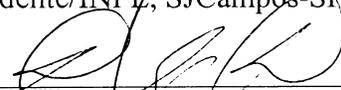
Aprovada pela Banca Examinadora em
cumprimento a requisito exigido para a
obtenção do Título de **Mestre em
Computação Aplicada.**

Dr. José Demisio Simões da Silva



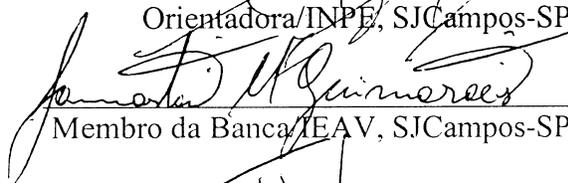
Presidente/INPE, SJCampos-SP

Dr^a Sandra Aparecida Sandri



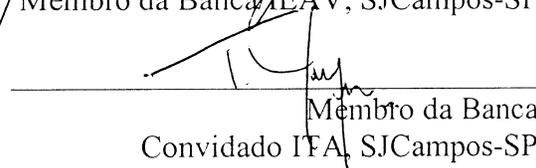
Orientadora/INPE, SJCampos-SP

Dr. Lamartine Nogueira Frutuoso Guimarães



Membro da Banca/FEAV, SJCampos-SP

Dr. Takashi Yoneyama



Membro da Banca
Convidado IFA, SJCampos-SP

Candidata: Isabela Neves Drummond

São José dos Campos, 26 de fevereiro de 2003.

“A mente que se abre a uma nova idéia jamais voltará ao seu tamanho original”.

ALBERT EINSTEIN

*Dedico este trabalho aos meus pais.
Que todo o meu esforço possa retribuir a dedicação
e a confiança que vocês sempre tiveram em mim.*

AGRADECIMENTOS

Agradeço a Deus.

Agradeço à minha orientadora, Dra. Sandra Aparecida Sandri, pelo apoio, incentivo e orientação segura e amiga ao longo deste trabalho.

Ao professor Dr. José Demísio Simões da Silva pela amizade e pela atenção dispensada a todas as minhas dúvidas.

Ao professor Dr. Lluís Godó pelo apoio e contribuição no desenvolvimento deste trabalho e pela disposição em responder todas as minhas mensagens.

Ao professor Dr. Takashi Yoneyama pela oportunidade e orientação no estágio de docência realizado no Instituto Tecnológico de Aeronáutica (ITA).

Aos membros da banca examinadora pela disposição em analisar este trabalho.

Ao meu pai pelo incentivo e por acreditar em mim sempre.

À minha mãe, pelo amor e pela dedicação, e ao meu irmão que sempre torceu por mim.

Ao meu namorado Alexandre, quem eu tanto admiro, pelo amor, carinho, atenção, assistência, paciência e incentivo em tudo o que faço.

Às minhas eternas amigas Bet e Cris por estarem sempre presentes, mesmo tão longe.

Aos meus amigos Ana Paula e Elcio pela amizade sincera e, principalmente, por estarem sempre ao meu lado.

Aos amigos Leo e Fabrício, pela amizade, companheirismo e apoio nesta jornada.

Às meninas lá de casa, Marizete e Karinere.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo auxílio financeiro, ao Instituto Nacional de Pesquisas Espaciais (INPE) pela oportunidade e apoio e a todos os professores e funcionários do Laboratório Associado de Computação e Matemática Aplicada (LAC), em especial, às secretárias Cristina e Tâmara por todos os favores prestados.

Agradeço a todos aqueles que direta e indiretamente colaboraram para a realização deste trabalho.

RESUMO

Este trabalho apresenta a utilização de relações de similaridade para tratar problemas de inconsistência e de falta de cobertura que podem ser verificados em uma base de conhecimento com regras do tipo “Se X é A_i então Y é B_i ”, onde A_i e B_i são conjuntos difusos. O foco do trabalho é em sistemas de regras difusas do tipo implicativo, ou seja, a implementação do operador *se-então* é feita por operadores implicativos, e uma t-norma (especificamente o *min*) é utilizada para agregar as saídas. A abordagem baseada em similaridade consiste no emprego de relações de similaridade para aumentar a imprecisão dos conjuntos difusos utilizados numa dada aplicação. Desta maneira, é possível sanar inconsistências que ocorrem quando, para uma entrada válida, são ativadas regras conflitantes, e/ou problemas de falta de cobertura, que, por sua vez, ocorrem quando, para uma entrada válida, não existem regras cujas premissas enderecem esta entrada. Diferentes estratégias são definidas: a abordagem global, onde encontra-se uma relação de similaridade que é aplicada a toda a base de regras, e uma abordagem local, onde a relação de similaridade é aplicada a toda a base de regras, porém esta base modificada é utilizada apenas para as entradas que ativam regras conflitantes. Neste trabalho, verificam-se os requisitos de uma relação de similaridade cuja aplicação leve a uma perda mínima de informação. Um algoritmo genético é empregado para aprender os parâmetros que definem os termos difusos e as relações de similaridade para uma determinada aplicação, visando a otimização do desempenho de um dado sistema. O trabalho traz uma análise comparativa entre o desempenho de um controlador difuso conjuntivo, do tipo Mamdani, e o desempenho de um controlador implicativo (usando os operadores de Rescher-Gaines, Gödel e Goguen), no qual foi aplicada a abordagem proposta.

THE USE OF SIMILARITY RELATIONS TO DEAL WITH CONSISTENCY AND COVERING PROBLEMS IN SYSTEMS OF FUZZY RULES

ABSTRACT

This work presents the use of similarity relations to deal with inconsistency and covering problems that may occur in a knowledge base with rules of the type “If X is A_i then Y is B_i ”, where A_i and B_i are fuzzy sets. The work is focused on systems using implicative fuzzy rules based systems, i.e, in which the *if-then* operator is implemented by a truly implication operator, and a t-norm (the min operator) is employed to aggregate the output. The approach based on similarity consists of the use of similarity relations to enlarge the imprecision of the fuzzy sets employed in a given application. On this way, it is possible to solve inconsistencies, that occur when, for a valid input, the fired rules present conflict, and/or covering problems, that occur when, for a valid input, there are no rules whose premises address the input. Different strategies are defined: the global approach, where a similarity relation is applied to all the fuzzy terms, and the local approach, where a similarity relation is applied to all the terms, but the modified terms are used only for the inputs which fired conflicting rules. In this work, the constraints that a similarity relation must obey so that its application induces a minimum loss of information. A genetic algorithm is also employed in order to learn the best parameters that define the fuzzy terms and the similarity relations for a given application, aiming at an optimization of the system performance. This work also brings a comparative analysis between the performance of a conjunctive fuzzy controller of the Mamdani kind and the performance of an implicative controller (employing Rescher-Gaines, Gödel and Goguen operators), on which the proposal approach was applied.

SUMÁRIO

	<u>Pág.</u>
LISTA DE FIGURAS	15
LISTA DE TABELAS	17
CAPÍTULO 1 – INTRODUÇÃO	19
CAPÍTULO 2 – ELEMENTOS DA TEORIA DOS CONJUNTOS DIFUSOS	23
2.1 – Termos básicos	23
2.2 – Operações	25
2.3 – Operadores de implicação	26
2.4 – Relações difusas	27
2.5 – Relações de similaridade	28
CAPÍTULO 3 – SISTEMAS BASEADOS EM REGRAS DIFUSAS	31
3.1 – Regras difusas	31
3.2 – Variáveis lingüísticas	32
3.3 – Mecanismo de inferência	33
3.4 – Interfaces de codificação e decodificação	34
3.5 – Controladores difusos	35
3.6 – O sistema <i>shower</i>	36
CAPÍTULO 4 – CONSISTÊNCIA E COBERTURA	39
4.1 – O problema da inconsistência	39
4.1.1 – A consistência em um sistema implicativo	39
4.1.2 – Abordagem baseada em relações de similaridade	40
4.1.3 – Família de relações de similaridade	43
4.1.4 – Estratégias de aplicação da abordagem baseada em similaridade	45
4.2 – O problema da falta de cobertura	46
CAPÍTULO 5 – DETERMINAÇÃO DE RELAÇÕES DE SIMILARIDADE POR PERDA MÍNIMA DE INFORMAÇÃO	51
5.1 – Verificação de inconsistências	51
5.2 – Cálculo do valor de λ	53

5.3 – Experimentos	54
5.4 – Resultados	55
5.5 – Desempenho dos controladores difusos	56
5.6 – Emprego do operador de Gödel	57
CAPÍTULO 6 – DETERMINAÇÃO DE RELAÇÕES DE SIMILARIDADE POR APRENDIZADO	63
6.1 – Algoritmos genéticos	63
6.2 – Aprendizado de termos difusos	64
6.3 – Características do AG implementado	65
6.4 – Experimentos	67
CAPÍTULO 7 – CONCLUSÕES E TRABALHOS FUTUROS	73
REFERÊNCIAS BIBLIOGRÁFICAS	77

LISTA DE FIGURAS

	<u>Pág.</u>
2.1 Cardinalidade, altura, suporte, núcleo e corte de nível 0.5 do conjunto difuso A	25
3.1 Variável lingüística temperatura.	32
3.2 Resultado da implicação $A \rightarrow B$ para os mecanismos de inferência de Mamdani, Gödel, Rescher-Gaines e Goguen.	34
3.3 Simulador sistema shower.	37
3.4 Partição do espaço de entrada: a) variável e_temp , b) variável e_flow	37
3.5 Partição do espaço de saída para as variáveis $cold$ e hot	37
4.1 Partição do espaço de saída.	45
4.2 Termos modificados utilizando $S_{\lambda,1}$	46
4.3 Termos modificados utilizando uma estratégia global.	46
4.4 Termos modificados utilizando uma estratégia local.	47
4.5 Regiões não cobertas pelo espaço de entrada.	47
4.6 Regiões não cobertas pelas regras da base.	48
4.7 Termos extrapolados, cobrindo as regiões W_1 e V_1	48
4.8 Termos extrapolados, cobrindo as regiões W_0 e V_0	49
5.1 a) Espaço de saída modificado utilizando $S_{0,3}$, b) detalhe.	53
5.2 Superfícies para a variável de saída $cold$: a) Mamdani, b) Rescher-Gaines com S_0	55

5.3	Superfícies para a variável de saída <i>cold</i> : a) Rescher-Gaines com $S_{0.3}$ global, b) Rescher-Gaines com $S_{0.3}$ local.	56
5.4	Variável <i>temp</i> : a) Mamdani, b) Rescher-Gaines com S_0 , c) Rescher-Gaines com $S_{0.3}$ global, d) Rescher-Gaines com $S_{0.3}$ local.	57
5.5	Saída do controlador: a) Mamdani, b) Rescher-Gaines com S_0	58
5.6	Saída do controlador: a) Rescher-Gaines com $S_{0.3}$ global, b) Rescher-Gaines com $S_{0.3}$ local.	59
5.7	Superfícies para a variável de saída <i>cold</i> : a) Gödel com $S_{0.1}$ local, b) Gödel com $S_{0.3}$ local.	59
5.8	Variável <i>temp</i> : a) Gödel com $S_{0.1}$ local, b) Gödel com $S_{0.3}$ local.	60
5.9	Variável <i>flow</i> : a) Gödel com $S_{0.1}$ local, b) Gödel com $S_{0.3}$ local.	60
5.10	Saída do controlador: a) Gödel com $S_{0.1}$ local, b) Gödel com $S_{0.3}$ local.	61
6.1	Partição simétrica.	66
6.2	Gödel: a) variável <i>temp</i> , b) variável <i>flow</i> , c) saída do controle.	68
6.3	Gödel: termos de entrada: a) variável <i>e_temp</i> , b) variável <i>e_flow</i> , c) termos que definem as variáveis de saída.	69
6.4	Goguen: a) variável <i>temp</i> , b) variável <i>flow</i> , c) saída do controle.	70
6.5	Goguen: termos de entrada: a) variável <i>e_temp</i> , b) variável <i>e_flow</i> , c) termos que definem as variáveis de saída.	71
6.6	Mamdani: a) variável <i>temp</i> , b) variável <i>flow</i> , c) saída do controle.	71
6.7	Mamdani: termos de entrada: a) variável <i>e_temp</i> , b) variável <i>e_flow</i> , c) termos que definem as variáveis de saída.	72

LISTA DE TABELAS

	<u>Pág.</u>
2.1 Principais operadores de implicação.	27
2.2 Principais operadores de conjunção usados como implicação.	28
3.1 Base de regras.	38
5.1 Base de regras para a variável <i>cold</i>	52
5.2 Valores da função de avaliação	57
6.1 Melhores indivíduos	69

CAPÍTULO 1

INTRODUÇÃO

A lógica difusa¹ (Zadeh, 1965; Driankov et al., 1996; Sandri e Correa, 1999) é uma tecnologia de inteligência computacional que vem sendo empregada com sucesso em diversos sistemas industriais. Na área de controle de processos, esta tecnologia permite o desenvolvimento dos chamados controladores difusos (Lee, 1990; Driankov et al., 1996), que, ao contrário dos controladores convencionais em que o algoritmo de controle é descrito através de um modelo matemático, são utilizadas regras lógicas para descrever em uma rotina a experiência humana para controlar um processo.

Os sistemas difusos são sistemas baseados em regras do tipo “Se X é A_i então Y é B_i ”, onde A_i e B_i são conjuntos difusos. Estes sistemas podem ser classificados basicamente como conjuntivos ou implicativos (Dubois e Prade, 1996), dependendo do tipo de operador *se-então* que relaciona a premissa de uma regra à sua conclusão. Nos sistemas conjuntivos são utilizados operadores de conjunção (t-normas, como o min e o produto), e nos implicativos são utilizados operadores de implicação propriamente ditos (como os operadores de Rescher-Gaines, Gödel e Goguen). Estas classes de operadores induzem diferenças no comportamento dos sistemas que os utilizam, particularmente no que tange à consistência entre as regras.

A representação do conhecimento é uma questão relevante no desenvolvimento de uma aplicação, sendo de fundamental importância buscar a forma de representação mais próxima do real. As regras difusas devem ser empregadas buscando restringir um conjunto de possíveis soluções. Quanto maior o número de regras, mais informação se tem e mais preciso pode ser o resultado obtido, sendo maior o custo de processamento. O ideal é encontrar o menor conjunto de regras que permita à aplicação ter um desempenho no nível desejado.

Em (Godo e Sandri, 2002b), um sistema baseado em regras difusas é chamado de completo se as seguintes restrições forem satisfeitas: (i) os termos associados às variáveis de entrada cobrem todos os possíveis valores de entrada, sendo ativada pelo menos uma regra na base de regras para cada entrada apresentada; (ii) para qualquer entrada apresentada, o sistema é capaz de produzir uma saída diferente de um conjunto vazio, ou seja, o sistema é consistente.

¹O termo em inglês “fuzzy” é traduzido para o português como difuso ou nebuloso.

A maioria das aplicações de sistemas difusos na atualidade são baseadas em regras conjuntivas, como os trabalhos de Mamdani em controle difuso (Mamdani, 1974; Driankov et al., 1996). Um sistema conjuntivo, do ponto de vista da lógica, nunca apresenta inconsistências, não representando, porém, o significado real das regras em um sistema especialista mas, sim, um mapeamento (entrada, saída)(Dubois et al., 1996).

Um conjunto de regras baseadas em implicações residuais possibilita a construção de regras do tipo “Quanto mais X é A mais Y é B ”, chamadas regras difusas graduais. Do ponto de vista semântico, esta representação é, claramente, uma maneira melhor de descrever a informação usualmente encontrada em regras de produção. Contudo, inconsistências são muitas vezes verificadas em sistemas implicativos, ou seja, uma entrada dispara regras que apresentam conclusões conflitantes, o que talvez explique sua reduzida utilização na prática. No entanto, este problema pode ser solucionado através da utilização de relações de similaridade (Godo e Sandri, 2002b). Estas relações também podem ser usadas para garantir a cobertura da base, sanando um problema que pode ocorrer tanto em sistemas conjuntivos quanto implicativos (Godo e Sandri, 2002a).

O objetivo do trabalho é utilizar relações de similaridade para tratar consistência e cobertura em sistemas de regras difusas. Na solução dos problemas de inconsistência, esta abordagem substitui os termos conseqüentes das regras por termos mais imprecisos², obtidos através de uma relação de similaridade adequada. Para superar os problemas de falta de cobertura, modifica-se a premissa das regras envolvidas, tornando-as mais imprecisas. Para tratar ambos os casos, neste trabalho foram empregadas famílias de relações de similaridade que apresentam certas características interessantes. O trabalho visa também estudar o uso de técnicas de aprendizado que permitam encontrar os melhores parâmetros (das relações de similaridade) para uma dada aplicação.

Os procedimentos referentes ao projeto foram implementados através do ambiente ©MatLab, com a utilização dos pacotes relacionados ao problema (Roger e Gulley, 1995), incluindo o sistema de controle utilizado para realizar os testes. A bateria de testes realizada empregando diferentes configurações foi a base da análise de desempenho dos controladores tradicionais (que são baseados em sistemas conjuntivos) e dos sistemas implicativos.

Este documento é organizado da seguinte maneira:

²Dados dois conjuntos difusos A e B , A é dito ser maior (ou mais impreciso) ou igual que B se $A(x) \geq B(x)$.

- No Capítulo 2 são apresentadas algumas das características dos conjuntos difusos, enfocando particularmente os aspectos mais relevantes para este trabalho.
- O Capítulo 3 concentra-se nos sistemas baseados em regras. São apresentados os tipos de regras difusas que originam os sistemas conjuntivos e implicativos e a estrutura de tais sistemas formados por um mecanismo de inferência, uma base de conhecimento e pela interface de entrada e saída. Os controladores difusos são descritos neste capítulo, sendo apresentado em seguida um experimento que emprega controlador difuso.
- O Capítulo 4 refere-se ao emprego das relações de similaridade para tratar consistência e cobertura em uma base de regras. Primeiramente são definidos os problemas que podem ser identificados numa base de regras, descrevendo-se em seguida como utilizar as relações de similaridade na obtenção de sistemas difusos completos.
- No Capítulo 5 é apresentada a primeira abordagem empregada para se determinar relações de similaridade, tendo como objetivo a perda mínima de informação. São descritos os procedimentos implementados e, ao final, são feitas considerações sobre os experimentos realizados, com uma análise de desempenho dos controladores difusos conjuntivo e implicativo, aplicando a abordagem proposta.
- No Capítulo 6 é descrita a determinação de relações de similaridade por aprendizado. Esta abordagem emprega algoritmos genéticos para buscar os melhores parâmetros para uma determinada aplicação. É feita uma breve introdução sobre a ferramenta de busca empregada e, em seguida, são descritas as características do algoritmo implementado. Os resultados obtidos são apresentados e analisados considerando-se os diferentes controladores difusos (conjuntivos e implicativos).
- Ao final, no Capítulo 7, são apresentadas as principais conclusões e propostas de trabalhos futuros.

CAPÍTULO 2

ELEMENTOS DA TEORIA DOS CONJUNTOS DIFUSOS

O objetivo deste capítulo é apresentar conceitos dos conjuntos difusos e seus principais aspectos relevantes para este trabalho.

O conceito de conjunto difuso e algumas idéias básicas sobre operadores destes conjuntos foram primeiramente introduzidos por Lotfi Zadeh em 1965 (Zadeh, 1965). A teoria dos conjuntos difusos lida com informações de caráter vago ou impreciso (Tanaka, 1991; Tsoukalas e Uhrig, 1997; Pedrycz e Gomide, 1998).

Na teoria clássica dos conjuntos, um elemento pertence ou não a um determinado conjunto. Assim, a pertinência de um elemento particular x de um conjunto A em um determinado universo U pode ser definida através de uma função característica $A(x) : U \rightarrow \{0, 1\}$ ¹ onde $A(x) = 1$ se $x \in A$ e $A(x) = 0$ se $x \notin A$.

Já um conjunto difuso A , em um universo U , é definido por uma função de pertinência $A(x) : U \rightarrow [0, 1]$ mapeando os valores do domínio no intervalo dos reais em $[0, 1]$. A função de pertinência $A(x)$ indica o grau de compatibilidade entre x e o conceito expresso por A :

- $A(x) = 1$, x é completamente compatível com A ;
- $A(x) = 0$, x é completamente incompatível com A ;
- $0 < A(x) < 1$, x é parcialmente compatível com A , com grau $A(x)$;

A lógica difusa se apresenta como uma forma de representação da incerteza e imprecisão, podendo ser vista também como um paradigma de raciocínio aproximado (Tanaka, 1991; Yen, 1989). Apesar de polêmica desde seu surgimento, esta tecnologia apresenta um grande número de aplicações industriais desenvolvidas com sucesso. Baseada na teoria dos conjuntos difusos, ela é empregada como um sistema de conceitos, princípios e métodos que lidam com modos de raciocínio que se aproximam do raciocínio exato.

2.1 Termos básicos

Dado um conjunto difuso A definido em U , sua cardinalidade é expressa como:

¹Neste trabalho adota-se a mesma notação para caracterizar um conjunto e a função que a descreve (função característica no caso clássico e função de pertinência no caso difuso).

- Para o universo de discurso discreto:

$$|A| = \sum_{x \in U} A(x) \quad (2.1)$$

- Para o universo de discurso contínuo:

$$|A| = \int_U A(x) \quad (2.2)$$

A representação aproximada de um conjunto difuso A é obtida através de conjuntos clássicos² em U . Estes conjuntos são subconjuntos A_α , denominados de cortes de nível e definidos da seguinte maneira:

$$A_\alpha = \{x \in U / A(x) \geq \alpha\} \quad (2.3)$$

A partir deste conceito, podem ser definidos o suporte de A , $Su(A)$ e o núcleo de A , $Nu(A)$. O primeiro agrupa elementos de U que são de alguma forma compatíveis com o conceito expresso por A , e o segundo agrupa elementos de U que são completamente compatíveis com o conceito expresso por A . Logo, pode-se definir:

$$Su(A) = \{x \in U / A(x) > 0\} = \lim_{\alpha \rightarrow 0} A_\alpha \quad (2.4)$$

$$Nu(A) = \{x \in U / A(x) = 1\} = A_1 \quad (2.5)$$

A altura de A definida como o supremo da função $A(x)$:

$$Al(A) = \sup_{x \in U} A(x) \quad (2.6)$$

Se $Al(A) = 1$, tem-se um conjunto difuso normalizado. As definições apresentadas estão ilustradas na Figura 2.1(Sandri e Correa, 1999).

²Aqui os conjuntos convencionais da teoria dos conjuntos são chamados de clássicos.

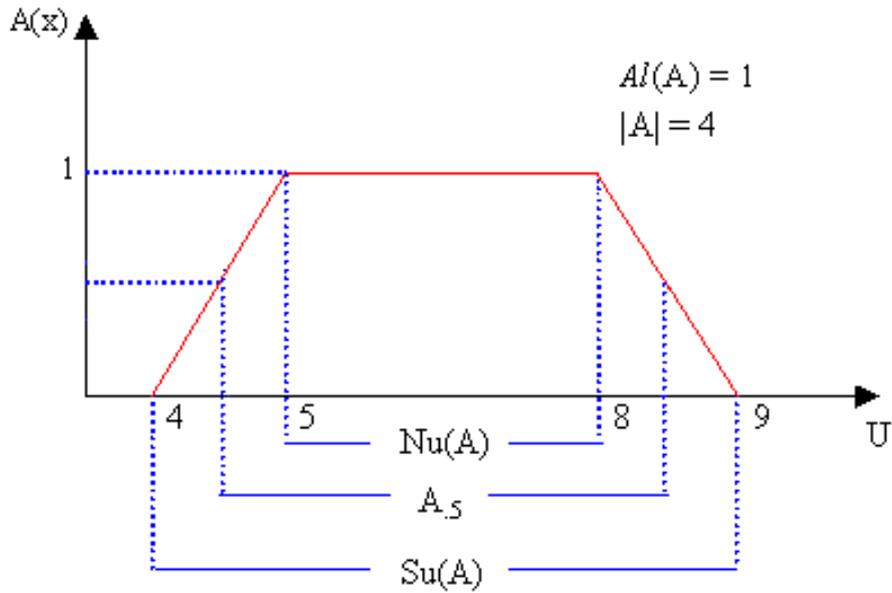


FIGURA 2.1 – Cardinalidade, altura, suporte, núcleo e corte de nível 0.5 do conjunto difuso A .

Um conjunto difuso A é *convexo* (Driankov et al., 1996) se, e somente se,

$$\forall x, y \in X \quad \forall \lambda \in [0, 1] : A(\lambda \cdot x + (1 - \lambda) \cdot y) \geq \min(A(x), A(y)) \quad (2.7)$$

Como exemplo de aplicação, os conjuntos convexos são normalmente utilizados para representar números difusos.

Na maior parte das aplicações, um conjunto difuso A pode ser representado utilizando a notação $\langle a, b, c, d \rangle$, onde $Su(A) = [a, d]$ e $Nu(A) = [b, c]$ e as funções entre a e b e d e c são estritamente monotônicas (crescente e decrescente, respectivamente). Este tipo de conjunto será chamado aqui de *conjunto difuso padrão*. Quando a função de pertinência é linear por partes, o conjunto difuso é chamado de *trapezoidal*, como o conjunto representado na Figura 2.1, onde $\langle a, b, c, d \rangle = \langle 4, 5, 8, 9 \rangle$. Um conjunto difuso triangular é denotado simplificada por $\langle a, b, d \rangle = \langle a, b, b, d \rangle$. Neste trabalho, os limites esquerdo e direito de um intervalo $[a, b]$ são denotados por $E([a, b]) = a$ e $D([a, b]) = b$

2.2 Operações

A conjunção e a disjunção de conjuntos difusos são obtidas utilizando-se os conceitos de t-norma e t-conorma, respectivamente. As t-normas constituem uma família de

operadores $\top : [0, 1] \times [0, 1] \rightarrow [0, 1]$ tais que, para quaisquer $x, y, z, w \in [0, 1]$, \top satisfaz as seguintes propriedades:

- **Comutatividade:** $x \top y = y \top x$
- **Associatividade:** $(x \top y) \top z = x \top (y \top z)$
- **Monotonicidade:** $x \top w \leq y \top z$, se $x \leq y, w \leq z$
- **Elemento neutro = 1:** $x \top 1 = x$

As t-conormas \perp , da mesma forma, são funções $\perp : [0, 1] \times [0, 1] \rightarrow [0, 1]$ comutativas, associativas e monotônicas. A diferença é que neste caso o elemento neutro é igual a 0, ou seja $x \perp 0 = x$. Em consequência destas propriedades tem-se $x \top 0 = 0$ e $x \perp 1 = 1$ (elemento absorvente).

Estes operadores são utilizados para implementar a união e a intersecção de conjuntos difusos. Por exemplo, pode-se fazer a união de dois conjuntos A e B utilizando a t-conorma max, $A \cup B(x) = \max(A(x), B(x))$ e a intersecção utilizando a t-norma min, $A \cap B(x) = \min(A(x), B(x))$.

A operação complemento também pode ser definida sobre os conjuntos difusos. A função $N : [0, 1] \rightarrow [0, 1]$ é chamada negação se é uma função bijetora decrescente, contínua e tal que $N(0) = 1$ e $N(1) = 0$. O complemento é dito involutivo se $N(N(x)) = x$ para todo $x \in [0, 1]$, sendo o complemento mais importante dado por $n(x) = 1 - x$.

2.3 Operadores de implicação

As regras de inferência *se-então* são modeladas pelos operadores de implicação $I : [0, 1] \times [0, 1] \rightarrow [0, 1]$. A relação $R = A \rightarrow B$ é expressa da seguinte maneira:

$$R(x, y) = I(A(x), B(y)), x \in X, y \in Y \quad (2.8)$$

onde A e B são dados por $A : X \rightarrow [0, 1], B : Y \rightarrow [0, 1]$.

As implicações propriamente ditas podem ser classificadas em categorias, dentre elas as implicações S (strong), que são da forma $I_s(a, b) = (N(a) \perp b)$ e as implicações R (residuais), que são tais que $I_R(a, b) = \sup\{c \in [0, 1] / (a \top c) \leq b\}$, onde \top é uma t-norma, \perp é uma t-conorma e N é uma negação.

Os principais operadores de implicação são mostrados na Tabela 2.1. As implicações de Lukasiewicz, Kleene e Reichenbach são exemplos típicos de implicações S . Exemplos

TABELA 2.1 – Principais operadores de implicação.

<i>Operador</i>	<i>nome</i>
$\max(1-a,b)$	Kleene-Dienes
$\min(1-a+b,1)$	Lukasiewicz
$\begin{cases} 1, & \text{se } a \leq b \\ 0, & \text{senão} \end{cases}$	Rescher-Gaines
$\begin{cases} 1, & \text{se } a \leq b \\ b, & \text{senão} \end{cases}$	Gödel
$\begin{cases} 1, & \text{se } a \leq b \\ b/a, & \text{senão} \end{cases}$	Goguen
$1-a+ab$	Reichenbach
$\max(1-a,\min(a,b))$	Zadeh-Wilmott

bem conhecidos de implicações R são Gödel e Goguen. O operador de Rescher-Gaines não está inserido na classe das implicações R , sendo obtido como resultado da intersecção de todos os operadores residuais (Bouchon-Meunier et al., 1999). A Tabela 2.2 apresenta os operadores de conjunção que são usualmente empregados como operadores de implicação³.

Muitas aplicações em lógica difusa (por exemplo, os controladores de Mamdani) utilizam uma t-norma para modelar o *se-então* nas regras. No entanto, estas funções não são implicações propriamente ditas, pois não correspondem à implicação da lógica booleana quando reduzidas, no caso da teoria dos conjuntos clássicos.

2.4 Relações difusas

As relações difusas (Dubois et al., 1998) são conjuntos difusos definidos em um universo multidimensional ($X_1 \times \dots \times X_n$). Como o próprio nome indica, uma relação implica na presença ou ausência de associação entre elementos de diferentes conjuntos. Na lógica difusa, o valor da função de pertinência das relações difusas indica o grau de associação entre os elementos.

Uma relação difusa R pode ser definida como um conjunto difuso, onde a cada ênupla (x_1, \dots, x_n) está associado um grau de pertinência entre 0 e 1. R é uma relação binária se definida em $X_1 \times X_2$. Tais relações são classificadas com base nas propriedades que apresentam.

³Ao longo do texto não é feita distinção entre os operadores de conjunção e os operadores de implicação propriamente ditos, a não ser quando necessário.

TABELA 2.2 – Principais operadores de conjunção usados como implicação.

<i>Operador</i>	<i>nome</i>
$\min(a,b)$	Mamdani
$a.b$	Larsen

Se R_1 e R_2 são relações difusas, respectivamente em $X \times Y$ e em $Y \times Z$, a composição de R_1 e R_2 , denotada por $R_1 \circ R_2$, resulta em uma nova relação que associa diretamente X a Z . A composição $sup\star$ pode ser definida da seguinte forma:

$$R_1 \circ R_2(x, z) = \sup_{y \in Y} [R_1(x, y) \odot R_2(y, z)] \quad (2.9)$$

onde $x \in X$, $z \in Z$ e \star é uma t-norma. A composição mais usual é a $sup-min$.

2.5 Relações de similaridade

As relações de similaridade, um tipo particular de relações difusas, foram originalmente introduzidas por Zadeh (1971) como uma generalização da definição clássica de relações de equivalência.

Basicamente, pode-se dizer que uma relação binária difusa S no universo U é uma relação de similaridade (Ruspini et al., 1998; Dubois et al., 1998) em U se as seguintes propriedades forem satisfeitas:

$$S(x, x) = 1 \quad \text{reflexividade} \quad (1)$$

$$S(x, y) = S(y, x) \quad \text{simetria} \quad (2)$$

$$S(x, y) \odot S(y, z) \leq S(x, z) \quad \text{transitividade} \quad (3)$$

para todo $x, y, z \in X$, onde \odot é uma t-norma.

Na literatura, freqüentemente, a transitividade não é requerida e, neste caso, quando somente a reflexividade e a simetria são obedecidas, a relação é chamada *relação de proximidade*.

Sejam S_1 e S_2 duas relações de similaridade em um domínio U . S_1 é dita ser maior ou igual que S_2 se, para qualquer a, b em U , $S_1(a, b) \geq S_2(a, b)$.

Tem-se que um conjunto de relações de similaridade em U formam um reticulado ordenado. O topo deste reticulado é a maior similaridade, S_{\top} , que faz todos os elementos

serem completamente similares: $S_{\top}(v, v') = 1$ para todo $v, v' \in U$, não sendo possível distinguir um elemento do outro. A base deste reticulado S_{\perp} compreende a clássica relação de igualdade: $S_{\perp}(v, v') = 1$ se $v = v'$, senão, $S_{\perp}(v, v') = 0$, ou seja, é a relação que faz com que um elemento tenha grau de similaridade 1 consigo mesmo e 0 com os demais elementos (Dubois et al., 1998).

CAPÍTULO 3

SISTEMAS BASEADOS EM REGRAS DIFUSAS

Um sistema baseado em regras pode utilizar um conjunto de regras do tipo *se-então* e uma coleção de fatos para inferir novas informações. As regras constituem uma maneira formal de representação das diretivas e estratégias que descrevem o comportamento de um sistema.

Nos sistemas difusos baseados em regras, a informação representada pelas regras difusas pode ser entendida da seguinte maneira: dado um fato, “ X é A ”, e uma regra “Se X é A então Y é B ”, onde A , A' e B são conjuntos difusos, com A' não necessariamente igual a A , deseja-se obter um novo fato, “ Y é B' ”, como uma consequência lógica (Driankov et al., 1996; Pedrycz e Gomide, 1998; Ruspini et al., 1998), tal que B' será mais próximo de B quanto mais A' for próximo de A .

Diferentemente da lógica clássica, onde a inferência pode ser obtida através da manipulação de símbolos, o processo de inferência na lógica difusa é obtido através da manipulação numérica das funções de pertinência dos conjuntos difusos.

Usualmente, os sistemas baseados em regras são formados por um mecanismo de inferência, uma base de conhecimento e pela interface de entrada e saída. No caso dos sistemas difusos, muitas vezes nas interfaces de entrada e saída são encontrados um codificador e um decodificador.

3.1 Regras difusas

As regras difusas (Dubois e Prade, 1996) são empregadas como uma forma de expressar o conhecimento na lógica difusa, ou seja, consistem em um esquema para capturar o conhecimento que envolve imprecisão. As regras difusas encontradas mais freqüentemente na literatura podem ser classificadas em regras conjuntivas ou regras implicativas (Dubois e Prade, 1996; Ruspini et al., 1998; Yen, 1989), em função do tipo do operador *se-então* empregado na implementação¹.

As regras conjuntivas fazem uso de uma t-norma, ou seja, um operador de conjunção, como min ou o produto, para implementar o operador *se-então*. A agregação das saídas obtidas é implementada por um operador de disjunção (t-conorma), normalmente o operador max.

¹Outros tipos de regras, como aquelas empregadas em controladores do tipo Takagi-Sugeno (Takagi e Sugeno, 1985) ou ainda em outras abordagens (Yu e Bien, 1994), não são consideradas neste trabalho.

As regras implicativas empregam operadores de implicação propriamente ditos, em vez de operadores de conjunção, para implementar o operador *se-então*, e uma t-norma para agregar as saídas, geralmente o operador min.

No modelo baseado em implicação, podem ser empregadas, por exemplo, as implicações *S* e as implicações *R*. No primeiro caso as regras obtidas correspondem semanticamente a declarações da forma “Quanto mais *x* é *A*, maior o grau de certeza de que *y* é *B*”, chamadas regras de certeza. O emprego do segundo tipo de implicação, as implicações residuais, geram as regras graduais da forma “Quanto mais *x* é *A*, mais *y* é *B*”.

Em (Dubois e Prade, 1992, 1996), além das regras implicativas, as regras conjuntivas são classificadas como regras de possibilidade, que correspondem a declarações da forma “Quanto mais *x* é *A*, maior a possibilidade de *B*” e como regras antigraduais, que correspondem a declarações do tipo “Quanto mais *x* é *A* e menos *y* está relacionado com *x*, menos *y* é *B*”.

3.2 Variáveis lingüísticas

Uma variável lingüística é aquela cujos valores possíveis são conjuntos difusos. Formalmente, ela pode ser definida como uma tripla $(X, U, T(X))$, onde *X* é uma variável, *U* é o universo de discurso de *X* e $T(X)$ é um conjunto de conjuntos difusos em *X* chamados de termos lingüísticos.

Por exemplo, a variável lingüística *temperatura*, representada na figura 3.1, pode assumir como valores um dos termos do conjunto $\{baixa, média, alta\}$. O grau com que um valor *x* em *U* satisfaz o termo lingüístico *A* é a pertinência de *x* em *A* dada por $A(x)$.

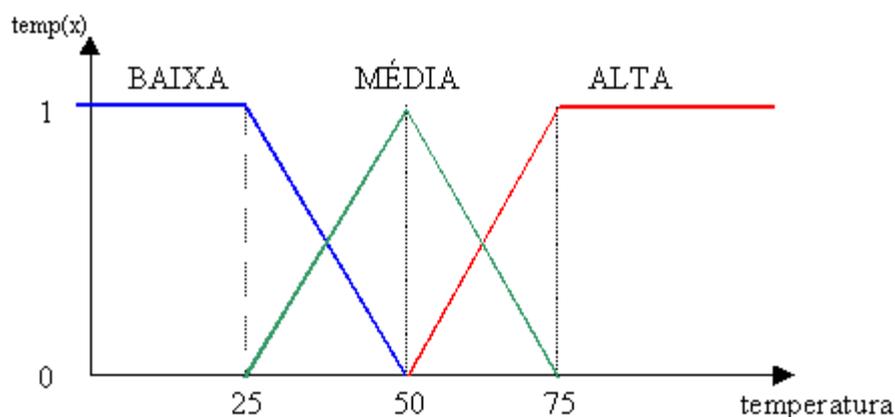


FIGURA 3.1 – Variável lingüística temperatura.

Um caso particular de variável lingüística é aquele em que os termos difusos que definem esta variável formam a chamada partição de Ruspini (Ruspini, 1969). Dado que um conjunto de termos $\{T_1, \dots, T_n\}$ está em ordem consecutiva quando, para todo i , se $Su(T_i) \cap Su(T_{i-1}) \neq \emptyset$ e $Su(T_i) \cap Su(T_{i+1}) \neq \emptyset$ então $Su(T_i) \cap Su(T_j) = \emptyset, \forall j \notin \{i-1, i+1\}$. Em uma partição de Ruspini, todos os pares de termos consecutivos T e T' são tais que $T(\omega) + T'(\omega) = 1$ para todo $\omega \in Su(T) \cap Su(T')$. Ou seja, informalmente, na partição de Ruspini o núcleo do termo T acaba onde começa o suporte do termo consecutivo T' .

3.3 Mecanismo de inferência

O mecanismo de inferência difuso utiliza os princípios da lógica difusa para determinar como os fatos e as regras devem ser combinados para derivar novos fatos.

A inferência pode ser baseada em composição ou no disparo individual de regras. No primeiro caso, dada uma entrada qualquer, ocorre o disparo de um conjunto de regras cujas relações estão compostas em uma única relação (as relações individuais podem ser compostas utilizando, por exemplo, uma união ou uma intersecção). No segundo caso, mais encontrado na prática, cada regra determina um conjunto difuso de saída e a saída total do mecanismo de inferência é a combinação dos conjuntos difusos individuais, feita usualmente por união ou por intersecção.

Diferentes mecanismos de inferência são propostos na literatura e bastante utilizados em aplicações reais. Dentre eles, destaca-se a inferência do tipo Mamdani (Mamdani, 1974), onde a implicação nas regras é modelada pelo operador min e a agregação dos resultados é feita pelo operador max. Sistemas que utilizam um operador de intersecção para modelar a implicação, como é o caso de Mamdani, são chamados de conjuntivos e os sistemas que usam operadores de implicação propriamente ditos, como os operadores de Rescher-Gaines, de Gödel e de Goguen, são chamados de implicativos.

Pode-se observar na Figura 3.2 o resultado da inferência para o caso de uma regra “Se X é A então Y é B ” disparada para uma dada entrada. Esta figura compara a saída das funções de pertinência para quatro tipos de implicação: Mamdani, Gödel, Rescher-Gaines e Goguen (vide Tabelas 2.1 e 2.2).

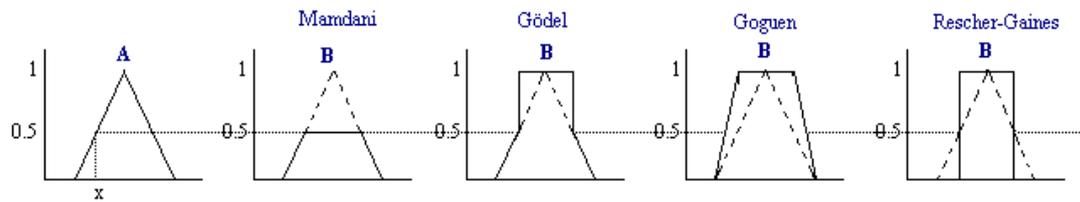


FIGURA 3.2 – Resultado da implicação $A \rightarrow B$ para os mecanismos de inferência de Mamdani, Gödel, Rescher-Gaines e Goguen.

3.4 Interfaces de codificação e decodificação

A interface entre o mecanismo de inferência, que trabalha com variáveis lingüísticas, e o ambiente de uma dada aplicação usualmente emprega os chamados mecanismos de codificação² e de decodificação.

A decodificação produz uma saída precisa (um único valor) a partir do conjunto difuso obtido pelo sistema de inferência. Dentre os métodos de decodificação (Driankov et al., 1996; Sandri e Correa, 1999) utilizados, destacam-se três: o primeiro máximo (SOM - smallest of maximum), a média dos máximos (MOM - mean of maximum) e o centro de área (COA - center of area). O método SOM examina o conjunto difuso de saída e determina, como valor de saída, o valor do ponto em que o grau de pertinência atinge o primeiro valor máximo. O método MOM examina o conjunto difuso de saída e, primeiro, determina os pontos em que o grau de pertinência é máximo. A saída é o valor obtido a partir da média dos valores de tais pontos. No método COA o valor de saída é o centro de gravidade do conjunto difuso de saída. Quando a saída do sistema é um conjunto clássico (vide seção 2.1), como ocorre quando o operador de Rescher-Gaines é utilizado, os métodos MOM e COA coincidem.

A codificação é empregada para transformar uma entrada precisa em um conjunto difuso. No entanto, a maior parte dos sistemas utiliza a entrada precisa diretamente no processamento. Na codificação também são feitas normalizações de domínios, se necessário. Por exemplo em muitas aplicações, o domínio de uma variável **erro** definida em $[-k, k] \subset \mathbb{R}$ é normalizado para $[-1, 1]$, com conseqüentes mudanças nos termos lingüísticos associados.

²Os termos codificação e decodificação são apresentados neste documento referindo-se aos termos em inglês “fuzzification” e “defuzzification”, respectivamente (Pedrycz e Gomide, 1998). Na literatura é comum a ocorrência de *fuzificação* e *defuzificação*.

3.5 Controladores difusos

Os controladores difusos (Lee, 1990; Driankov et al., 1996; Sandri e Correa, 1999) fornecem uma variedade de métodos que, em conjunto com técnicas tradicionais, podem resolver os modernos problemas de controle. Tais métodos devem apresentar características tais como: boa performance e robustez suficiente para lidar com perturbações e incertezas, dentre outros aspectos que envolvem a teoria de controle.

Os controladores convencionais apresentam algoritmos baseados num modelo matemático descrito por equações algébricas ou diferenciais. Já um controlador difuso baseia-se no conhecimento modelado através de conjuntos difusos.

Nestes sistemas, a estratégia de controle é descrita pelas regras lingüísticas que conectam as situações com as ações a serem tomadas. As regras são expressas na forma de implicações lógicas *se-então*, representando uma relação $R : A \times B \rightarrow [0, 1]$ entre um ou mais antecedentes e um ou mais conseqüentes.

A configuração básica de um controlador difuso apresenta quatro componentes principais: a interface de codificação, a base de conhecimento, o procedimento de inferência e a interface de decodificação.

A interface de codificação identifica os valores das variáveis de entrada, que caracterizam o estado do sistema e as normaliza em um universo de discurso padronizado. A base de conhecimento armazena as definições sobre discretização e normalização dos universos de discurso e as definições das funções de pertinência dos termos difusos, além da base de regras que caracteriza a estratégia de controle. As regras, juntamente com os dados de entrada, são processadas pelo procedimento de inferência, o qual infere as ações de controle de acordo com o estado do sistema, aplicando o operador de implicação. E, por fim, o decodificador é responsável pela obtenção de uma única ação de controle precisa, a partir do conjunto difuso obtido no processo de inferência.

O objetivo do projeto de um controlador difuso pode ser assim definido: dado um modelo (heurístico ou analítico) do sistema físico a ser controlado e as especificações do comportamento desejado, desenvolver um controlador empregando regras difusas que conduzam ao atendimento destas especificações.

O comportamento de um controlador difuso depende de inúmeros aspectos relacionados à sua estrutura e implementação. A construção do controlador envolve a definição do modelo e das características operacionais, para estabelecer as particularidades da arquitetura do sistema e a definição das propriedades operacionais do controlador.

Deve-se considerar a definição dos termos difusos de cada variável, permitindo que haja sobreposição parcial entre os conjuntos vizinhos, o que pode garantir suavidade e estabilidade. E, além disso, a construção do controlador envolve ainda a definição do comportamento do controle, onde são descritas as regras que atrelam as variáveis de entrada às propriedades de saída do modelo.

O projeto de um controlador envolve ainda o ajuste de parâmetros tais como o número de variáveis de entrada e saída, as funções de pertinência, a estrutura da base de regras, o universo de discurso e os parâmetros das funções de pertinência (como o núcleo e o suporte).

Em (Driankov et al., 1996) encontra-se uma lista de propriedades da base de regras que devem ser testadas, como cobertura, consistência, interação e robustez. A sintonização é uma tarefa complexa devido à flexibilidade que decorre da existência de muitos parâmetros, exigindo esforço do projetista na obtenção do melhor desempenho do controlador. Esta sintonização pode ser feita considerando-se três etapas. A primeira envolve o desenvolvimento de um controlador simples, que simule um controlador proporcional. Na segunda etapa, incrementa-se o conhecimento conforme a experiência resultante do processo, buscando-se novas variáveis lingüísticas ou físicas para contornar as dificuldades, ajustando-se as funções de pertinência e os parâmetros do controlador, adicionando-se regras, ou modificando a estrutura de controle. E, finalmente, é fundamental validar a consistência do conhecimento incorporado com novas condições de operação para o sistema.

Para o desempenho de tais tarefas é necessária uma plataforma sofisticada, com interface poderosa, e que permita uma rápida inferência. Para tanto, são disponibilizados pacotes dedicados à análise de modelos difusos.

3.6 O sistema *shower*

Nesta seção será apresentado o sistema *shower* que consiste em um controlador difuso disponível no ambiente de programação ©Matlab.

O experimento *shower* visa o controle da temperatura e da vazão da água de um chuveiro. A Figura 3.3 mostra, de maneira simplificada, a estrutura do controlador, onde as variáveis de saída do sistema são *temp* e *flow*. Para cada uma destas variáveis é definido um limiar, cujo valor deve ser atingido pelo sistema. Os valores de saída do sistema são comparados com os valores de referência, previamente definidos, gerando erros, *e_temp* e *e_flow*, para as variáveis, definindo assim as variáveis de entrada do controle difuso.

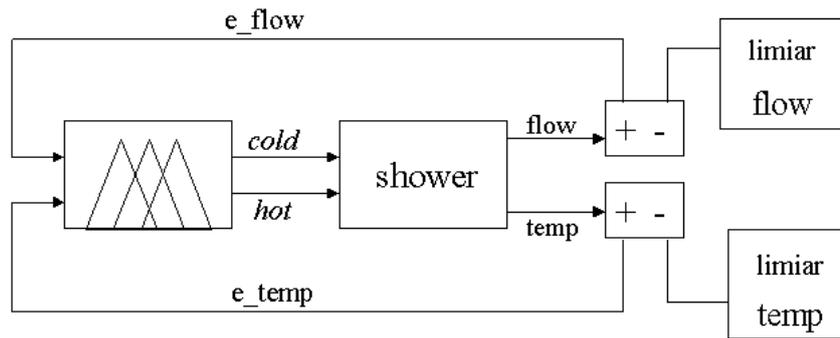


FIGURA 3.3 – Simulador sistema shower.

As variáveis de saída do controlador, *cold* e *hot*, determinam as ações de controle do sistema. Estas ações de controle são regidas por equações que não estão documentadas no aplicativo.

Detalhando o controle difuso no sistema *shower* tem-se as variáveis de entrada, *e_temp* e *e_flow* associadas aos conjuntos difusos $\{cold, good, hot\}$ e $\{soft, good, hard\}$, respectivamente (Figura 3.4). As duas variáveis de saída estão associadas aos termos difusos $\{close-fast(cf), close-slow(cs), steady(st), open-slow(os), open-fast(of)\}$, conforme pode ser visto na Figura 3.5. A base de regras, descrita na Tabela 3.1, é composta por nove regras do tipo “Se x_1 é A e x_2 é B então y é C ”, onde x_1 e x_2 representam os valores de cada uma das variáveis de entrada; y é a saída do sistema e A , B e C são conjuntos difusos.

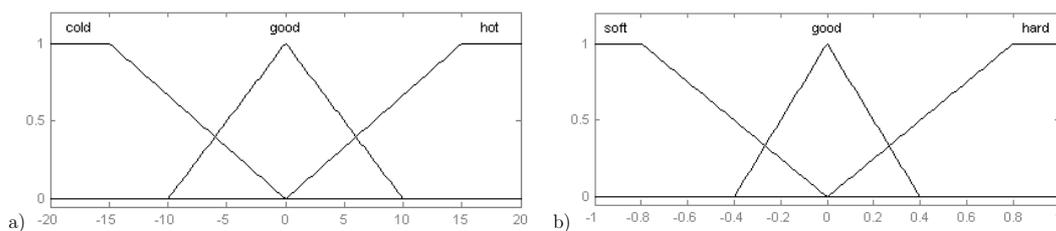


FIGURA 3.4 – Partição do espaço de entrada: a) variável *e_temp*, b) variável *e_flow*.

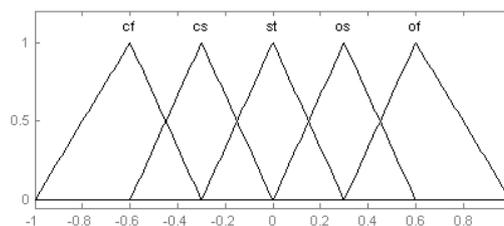


FIGURA 3.5 – Partição do espaço de saída para as variáveis *cold* e *hot*.

TABELA 3.1 – Base de regras.

(a) variável de saída *cold*

e_flow/e_temp	<i>cold</i>	<i>good</i>	<i>hot</i>
<i>soft</i>	<i>os (1)</i>	<i>os (4)</i>	<i>of (7)</i>
<i>good</i>	<i>cs (2)</i>	<i>st (5)</i>	<i>os (8)</i>
<i>hard</i>	<i>cf (3)</i>	<i>cs (6)</i>	<i>cs (9)</i>

(b) variável de saída *hot*

e_flow/e_temp	<i>cold</i>	<i>good</i>	<i>hot</i>
<i>soft</i>	<i>of (1)</i>	<i>os (4)</i>	<i>os (7)</i>
<i>good</i>	<i>os (2)</i>	<i>st (5)</i>	<i>cs (8)</i>
<i>hard</i>	<i>cs (3)</i>	<i>cs (6)</i>	<i>cf (9)</i>

CAPÍTULO 4

CONSISTÊNCIA E COBERTURA

Este trabalho baseia-se em duas propriedades que podem ser verificadas em uma base de regras de um sistema difuso: a cobertura e a consistência.

Considerando-se as características apresentadas para os sistemas difusos, pode-se dizer que um sistema cobre todo o espaço de entrada se, para cada entrada apresentada, é ativada, pelo menos, uma regra na base de regras. Além disso, um sistema é consistente se, para qualquer entrada válida apresentada, ele é capaz de produzir uma saída diferente de um conjunto vazio. Finalmente, define-se como um sistema difuso completo aquele que cobre todo o espaço de entrada e é consistente (Godo e Sandri, 2002b).

As próximas seções descrevem as situações em que um sistema difuso se apresenta incompleto e como os problemas verificados podem ser solucionados.

4.1 O problema da inconsistência

4.1.1 A consistência em um sistema implicativo

O problema da inconsistência é verificado apenas nos sistemas implicativos. Conforme descrito na seção 3.1, a agregação nos sistemas baseados em implicação é implementada por uma t-norma, geralmente o operador min. Assim, uma inconsistência pode ser facilmente detectada quando se verifica que uma entrada ativa regras cujos conseqüentes não apresentam intersecção.

Já um sistema conjuntivo nunca apresenta inconsistência, uma vez que a agregação das contribuições das várias regras ativadas para uma dada entrada é implementada através de uma t-conorma, ou seja, obtém-se como saída um conjunto não vazio (eventualmente não-convexo - vide seção 2.1) através da união dos conseqüentes das regras ativadas.

Uma solução para o problema da inconsistência é apresentada em (Godo e Sandri, 2002b), onde relações de similaridade (seção 2.5) são empregadas, transformando regras potencialmente inconsistentes em regras consistentes, viabilizando a aplicação dos sistemas implicativos.

Outras soluções para este problema podem ser delineadas, baseando-se em outros mecanismos que, de alguma maneira, tornem mais imprecisos os termos difusos. A vantagem de se utilizar a abordagem baseada em relações de similaridade, empregada

neste trabalho, é de que ela consiste em um método formal para solução do problema da inconsistência.

4.1.2 Abordagem baseada em relações de similaridade

A utilização de relações de similaridade para restaurar a consistência entre regras difusas baseadas em implicação é proposta em (Godo e Sandri, 2002b). A idéia desta abordagem é substituir os conseqüentes B_i 's das regras de produção “Se X é A_i então Y é B_i ” por termos mais imprecisos, através de uma relação de similaridade S adequada, de modo que se tenha como resultado regras de produção consistentes.

Emprega-se uma regra de transformação do tipo:

$$\frac{\text{Se } X \text{ é } A_i \text{ então } Y \text{ é } B_i}{\text{Se } X \text{ é } A_i \text{ então } Y \text{ é aproximadamente } _B_i} \quad (4.1)$$

que permite transformar um conjunto de regras de produção conflitantes em regras mais imprecisas, onde *aproximadamente* $_B_i$ é um conjunto difuso, maior que B_i , resultante da composição da relação de similaridade S com o conseqüente da i -ésima regra. O novo termo é definido como:

$$\text{aproximadamente_}B_i(v) = (S \circ B_i) = \sup_{v' \in U_Y} \min(S(v, v'), B_i(v')). \quad (4.2)$$

Quanto maior é a relação S , mais imprecisos são os conjuntos B_i^* obtidos ($B_i^* = S \circ B_i$) e mais imprecisas são as regras “Se X é A_i então Y é B_i^* ”.

Do ponto de vista da representação do conhecimento, é interessante que a perda de informação seja a menor possível quando, dado um sistema implicativo inconsistente, aplica-se a ele uma relação de similaridade. Por outro lado, deve-se assegurar que a relação de similaridade é suficientemente grande para eliminar as inconsistências.

É interessante analisar as soluções triviais para tratar a inconsistência, ou seja, qual o efeito de se tomar como solução do problema a maior relação de similaridade, $S = S_{\top}$ e a menor delas, $S = S_{\perp}$ (vide seção 2.5). A primeira garante que o sistema obtido será consistente mas, para qualquer entrada, a saída é totalmente irrestrita, ou seja, há uma perda completa de informação. Para a segunda solução obtém-se uma base de regras igual àquela apresentada para o sistema original, o que, por um lado, garante que

não houve perda de informação mas, por outro lado, faz com que o sistema permaneça inconsistente.

Assim, empregar a maior ou a menor relação de similaridade nesta abordagem não resolve o problema. É preciso que se determinem as condições necessárias para garantir consistência total em sistemas de regras difusas graduais sujeitos a algumas limitações, estabelecendo, desta forma, qual relação de similaridade, dentro de uma família de relações, deve ser empregada.

O conceito de consistência total está relacionado ao grau de consistência de um conjunto de regras $K = \{R_i\}_{i \in I}$, que é dado por

$$Cons(K) = \inf_x \sup_y \text{saida}(K, \{x\})(y) \quad (4.3)$$

onde a $\text{saida}(K, \{x\})(y)$ corresponde à saída do sistema quando a entrada $X = x$ é aplicada à base de regras K . Uma base de regras K é α -consistente, $\alpha \in [0, 1]$, se $Cons(K) = \alpha$. A consistência total, isto é, 1-consistência, é verificada quando $Cons(K) = 1$ (Dubois et al., 1996; Weisbrod e Fantana, 1996; Dubois et al., 1997).

Na abordagem que visa a perda mínima de informação, determina-se qual é a menor relação de similaridade que torna o sistema 1-consistente ((Drummond et al., 2002)). Embora esta solução seja ideal do ponto de vista desta abordagem, ela não garante que o desempenho do sistema em si seja melhor do que quando se utiliza uma relação maior que a mínima. Em (Drummond et al., 2003) emprega-se um algoritmo genético para aprender os parâmetros de relação de similaridade que otimizem o desempenho do sistema, mas garantindo sua α -consistência, $\alpha > 0$. Estas duas abordagens serão detalhadas nos próximos capítulos.

Para que seja determinada a relação de similaridade que será aplicada, independentemente da abordagem por informação mínima ou por aprendizado, o primeiro passo consiste na verificação da consistência do conjunto de regras que compõem a base do sistema considerado. Segundo Dubois, Prade e Ughetto (Dubois et al., 1996), este problema pode ser reduzido a verificar a consistência entre pares de regras.

Considerando que os termos difusos das regras $\{R_1, R_2\}$ são tais que A_1 vem antes de A_2 e B_1 vem antes de B_2 , ou seja, $D(Nu(A_1)) \leq E(Su(A_2))$ e $D(Su(A_1)) \leq E(Nu(A_2))$ e,

da mesma forma, para B_1 e B_2 , o par de regras considerado é 1-consistente se, e somente se ((Drummond et al., 2002)):

$$\begin{aligned} A_1(E(Su(A_2))) &\leq B_1(E(Su(B_2))), \\ A_2(D(Su(A_1))) &\leq B_2(D(Su(B_1))) \end{aligned} \quad (4.4)$$

A partir de 4.4 podem-se derivar as seguintes condições:

Seja $\delta = \max((A_1(E(Su(A_2))), A_2(D(Su(A_1))))$. Se $B_1 + B_2 \geq \delta$ na região definida pela intersecção $Su(B_1) \cap Su(B_2)$, então o par de regras $\{R_1, R_2\}$ é 1-consistente. Esta condição pode ser expressa como:

$$\begin{aligned} E(Su(B_2)) &\leq D(Su([B_1]_\delta)) \\ E(Su([B_2]_\delta)) &\leq D(Su(B_1)) \end{aligned} \quad (4.5)$$

Se A_1 e A_2 são tais que $A_1 + A_2$ são constantes (iguais a 2δ) em $Su(A_1) \cap Su(A_2)$, então o par de regras $\{R_1, R_2\}$ é 1-consistente se, e somente se, $B_1 + B_2 \geq \delta$ na região $Su(B_1) \cap Su(B_2)$.

Para o caso particular em que os termos difusos que definem a entrada formem uma partição de Ruspini, tem-se que $A_1 + A_2 = 1$ na região $Su(A_1) \cap Su(A_2)$ e $\delta = 1$, sendo as condições 4.5 reduzidas a:

$$\begin{aligned} E(Su(B_2)) &\leq D(Nu(B_1)) \\ E(Nu(B_2)) &\leq D(Su(B_1)) \end{aligned} \quad (4.6)$$

As condições apresentadas acima podem ser generalizadas para regras que apresentem n variáveis de entrada. Sejam as regras

$$R_i : \text{Se } x_1 \text{ é } A_i^1 \text{ e } x_2 \text{ é } A_i^2, \text{ então } y \text{ é } B_i, \text{ } i=1,2.$$

e as regras

$$R_i^1 : \text{Se } x_1 \text{ é } A_i^1 \text{ então } y \text{ é } B_i \quad R_i^2 : \text{Se } x_2 \text{ é } A_i^2 \text{ então } y \text{ é } B_i$$

R_1 e R_2 são 1-consistentes se, e somente se, os pares $\{R_1^1, R_2^1\}$ e $\{R_1^2, R_2^2\}$ são 1-consistentes. Deste modo, se as condições 4.5 forem satisfeitas para $\delta = \min(\delta_1, \delta_2)$ onde $\delta_j (j = 1, 2)$ é definido como anteriormente, mas em relação aos conjuntos A_1^j, A_2^j , então as regras apresentadas com duas variáveis de entrada são 1-consistentes.

No caso dos operadores residuais, quando a condição de 1-consistência é relaxada para α -consistência, $\alpha > 0$, a verificação da consistência é mais simples. Um par de regras $\{R_1, R_2\}$ é α -consistente se a seguinte restrição é satisfeita:

$$\text{Se } A_1 \cap A_2 \neq 0 \text{ então } B_1 \cap B_2 \neq 0$$

Com esta restrição garante-se que se uma dada entrada é ativada por duas regras $\{R_1, R_2\}$, então o suporte da intersecção de B_1 e B_2 é não vazio, embora possa ser subnormalizado.

4.1.3 Família de relações de similaridade

Para restaurar a consistência entre pares de regras utiliza-se aqui alguma família paramétrica $\mathcal{S} = \{S_0, S_{+\infty}\} \cup \{S_\lambda\}_{\lambda \in I \subseteq (0, +\infty)}$ de relações de similaridade “aninhadas” ((Godo e Sandri, 2002b; Drummond et al., 2002)), que seja tal que:

- a) $S_0 = S^\perp$,
- b) $S_{+\infty} = S^\top$ e
- c) $\lambda < \lambda'$, então $S_\lambda \prec S_{\lambda'}$

onde $S \prec S'$ significa $S(x, y) \leq S'(x, y)$ para todo $x, y \in U_Y$ e $S(x_0, y_0) < S'(x_0, y_0)$ para algum $x_0, y_0 \in U_Y$.

A seguir, serão descritas famílias de relações de similaridade que apresentam algumas características interessantes para o presente trabalho (Drummond et al., 2003).

Sejam dois conjuntos difusos $A = \langle a_1, a_2, a_3, a_4 \rangle$ e $B = \langle b_1, b_2, b_3, b_4 \rangle$, definidos em um universo U . A notação $A \ll B$, que corresponde a $a_i \leq b_i, i = \{1, 4\}$, indica que A está ordenado antes de B . Uma característica que as famílias de relações de similaridade devem apresentar, para serem utilizadas na abordagem apresentada neste trabalho, é serem tais que:

$$\text{Se } A \ll B \text{ então } S \circ A \ll S \circ B \tag{4.7}$$

Ou seja, aplicada uma relação de similaridade, a ordem dos conjuntos obtidos é preservada em relação aos originais.

As três classes de famílias de relações de similaridade que preservam a ordem e que serão utilizadas ao longo deste trabalho são definidas como se segue. Seja S uma relação de similaridade e $A = \langle a, b, c, d \rangle$, um conjunto difuso padrão em R^{+1} . Tem-se:

- Para $\lambda \geq 0$, $Su(S_\lambda \circ A) = [a - \lambda, d + \lambda]$;
- Para $\gamma \in (0, 1]$, $Su(S^\gamma \circ A) = [a \cdot \gamma, d/\gamma]$;
- Para $\lambda \geq 0$ e $\gamma \in (0, 1]$, $Su(S_{\lambda,\gamma} \circ A) = [a \cdot \gamma - \lambda, d/\gamma + \lambda]$, onde $S_{\lambda,\gamma} = S^\lambda \circ S_\gamma$;
- Para $\lambda \geq 0$ e $\gamma \in (0, 1]$, $Su(S^{\lambda,\gamma} \circ A) = [(a - \lambda)/\gamma, (d + \lambda)/\gamma]$, onde $S^{\lambda,\gamma} = S_\lambda \circ S^\gamma$;

As famílias S_λ e S^γ são consideradas casos particulares das classes representadas por $S_{\lambda,\gamma}$ e $S^{\lambda,\gamma}$: $S_{\lambda,1} = S^{\lambda,1} = S_\lambda$, e $S_{0,\gamma} = S^{0,\gamma} = S^\gamma$.

A família de relações $S_{\lambda,1}$ é tal que o suporte do conjunto A sofre um aumento igual a 2λ , não importando se o termo se encontra próximo de 0 ou não. Além disso, o mesmo valor de λ é aplicado aos dois lados do termo difuso. Assim, se A é simétrico, $A' = S_{\lambda,1} \circ A$ também será simétrico.

A classe $S_{0,\gamma}$ é tal que o suporte dos termos próximos de 0 sofrem um aumento menor do que os termos que se encontram no extremo do domínio; os lados dos termos difusos são aumentados a partir de diferentes constantes. Caso A seja simétrico, $A' = S_{0,\gamma} \circ A$ não será simétrico, exceto quando $\gamma = 1$. Para o caso em que é conhecido apenas um ponto por termo, o aprendizado desta relação de similaridade permite obter-se termos mais precisos na proximidade de 0. Entretanto, não é possível aplicar a relação $S_{0,\gamma}$ ao termo $A_0 = \langle 0, 0, 0, 0 \rangle$.

Um exemplo de uma relação de similaridade da classe S_λ é definido como se segue, para $U = \mathbb{R}$:

$$S_\lambda(x, y) = \max(0, 1 - \lambda^{-1} \cdot |x - y|) \quad (4.8)$$

onde $\lambda > 0$.

¹As relações de similaridade são aplicadas apenas aos termos difusos definidos no conjunto dos números reais positivos.

Para a classe S^γ pode-se definir a seguinte família de relações de similaridade, em \mathbb{R}^+ :

$$S^\gamma(x, y) = \begin{cases} 1, & \text{se } x = y = 0 \\ 0, & \text{se } \min(x, y)/\max(x, y) < \gamma \\ \min(x, y)/\max(x, y), & \text{senão} \end{cases} \quad (4.9)$$

onde $0 < \gamma \leq 1^2$.

As classes gerais, representadas por $S_{\lambda, \gamma}$ e $S^{\lambda, \gamma}$, com $(\lambda, \gamma) \neq (0, 1)$, combinam características das duas classes exemplificadas. Em particular, a classe $S^{\lambda, \gamma}$ é apropriada para se trabalhar com sistemas implicativos, uma vez que a constante multiplicativa pode ser utilizada para determinar o suporte dos termos difusos e a constante aditiva utilizada para eliminar as inconsistências.

A Figura 4.1 ilustra o conjunto de termos que definem uma variável de saída, chamada de partição de saída. Para exemplificar a aplicação de uma relação de similaridade, a Figura 4.2 apresenta a transformação dos termos Z e PM em Z^* e PM^* , utilizando a relação $S_{\lambda, 1}$.

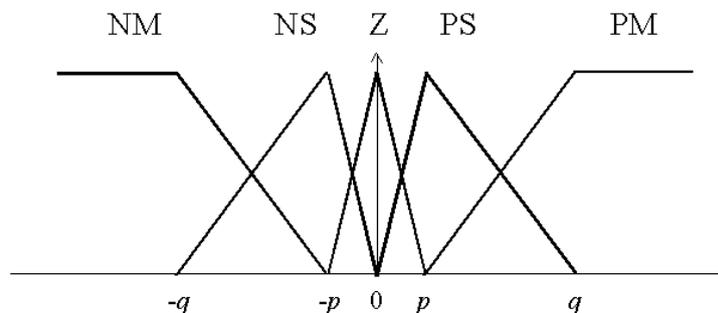


FIGURA 4.1 – Partição do espaço de saída.

4.1.4 Estratégias de aplicação da abordagem baseada em similaridade

As relações de similaridade podem ser utilizadas de diversas maneiras para restabelecer consistência, variando de uma abordagem global, onde uma relação de similaridade é aplicada a toda a base de regras, a uma abordagem completamente local, onde encontra-se uma relação de similaridade específica para cada entrada problemática verificada (Godo e Sandri, 2002b). Uma vez encontrada uma relação de similaridade

²Neste trabalho, não há interesse nos formatos específicos de B_i e $B_i^{*\gamma}$ (seção 4.1.2), mas somente nos seus respectivos núcleos e suportes.

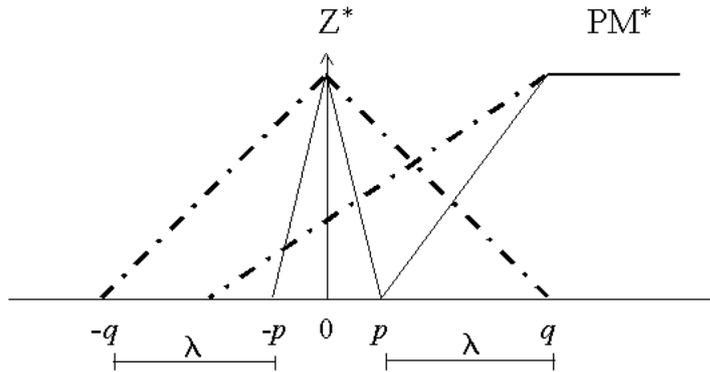


FIGURA 4.2 – Termos modificados utilizando $S_{\lambda,1}$.

adequada, ou seja, aquela que, quando aplicada, torna a base de regras consistente, é necessário optar pela estratégia que será empregada.

A Figura 4.3 mostra a partição de saída após a aplicação da relação de similaridade utilizando uma estratégia global, e a Figura 4.4 ilustra a aplicação da estratégia local considerando que as inconsistências somente ocorrem entre os termos Z e PM e entre Z e NM .

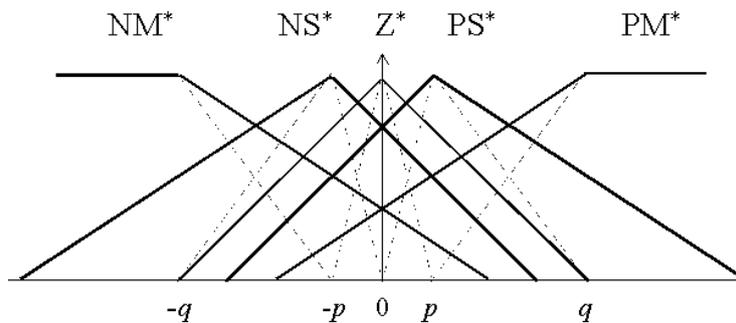


FIGURA 4.3 – Termos modificados utilizando uma estratégia global.

4.2 O problema da falta de cobertura

Tanto um sistema conjuntivo quanto um sistema implicativo podem apresentar problemas de falta de cobertura que ocorrem quando, para uma dada entrada, nenhuma regra da base é ativada. Duas situações são consideradas: a primeira ocorre quando a união dos suportes de todos os termos associados à variável de entrada não cobre todos os possíveis valores de entrada, ou seja, existem *buracos* na partição de entrada; a segunda situação é verificada quando o sistema apresenta mais de uma variável de entrada e, mesmo com o conjunto de termos cobrindo todo o espaço destas variáveis, para um vetor de entrada

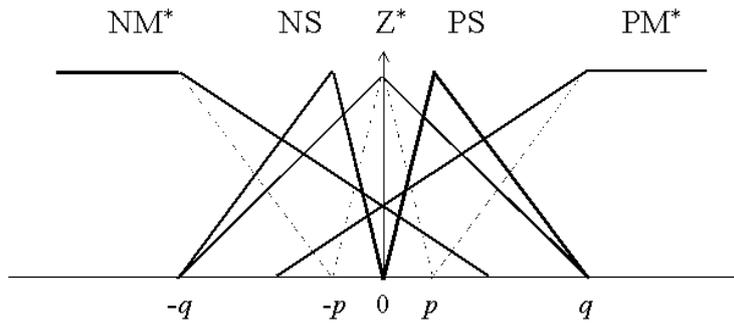


FIGURA 4.4 – Termos modificados utilizando uma estratégia local.

$X = \langle x_1, x_2, \dots, x_n \rangle$, não existe uma regra cuja premissa cubra o valor de X para todas as variáveis, simultaneamente.

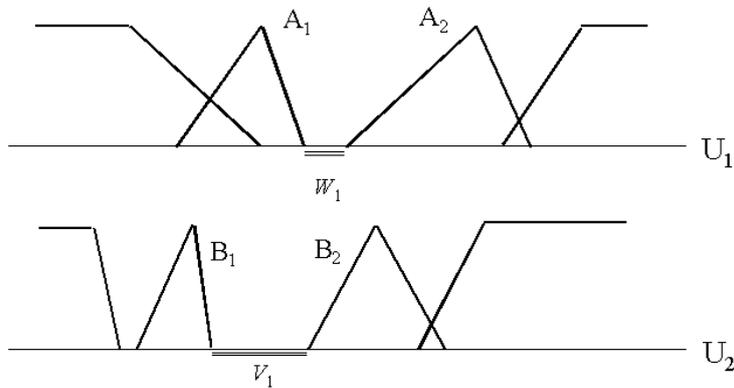


FIGURA 4.5 – Regiões não cobertas pelo espaço de entrada.

As duas situações podem ser observadas nas Figuras 4.5 e 4.6. No primeiro caso, as regiões W_1 e V_1 não são cobertas por nenhum dos termos que definem as variáveis.

Para o segundo caso, considere uma base de regras K composta pelas seguintes regras:

$$R_1: \text{Se } X_1 \text{ é } A_1 \text{ e } X_2 \text{ é } B_2 \text{ então } Y \text{ é } C_1$$

$$R_2: \text{Se } X_1 \text{ é } A_2 \text{ e } X_2 \text{ é } B_1 \text{ então } Y \text{ é } C_2$$

onde os domínios das variáveis X_1 e X_2 são $U_1 = U_2 = \mathbb{R}$, e cujos termos difusos que definem as variáveis estão representados na Figura 4.6. Pode-se observar que, apesar de não haver *buracos* nas partições, a região $\Omega = (W_0, V_0)$ não é coberta pela base de regras.

Uma solução empregada para sanar este problema é baseada em similaridade e apresentada em (Godo e Sandri, 2002a). Outros métodos para solucionar o problema da cobertura são propostos na literatura (Bouchon-Meunier et al., 2001) e se baseiam em mecanismos de interpolação.

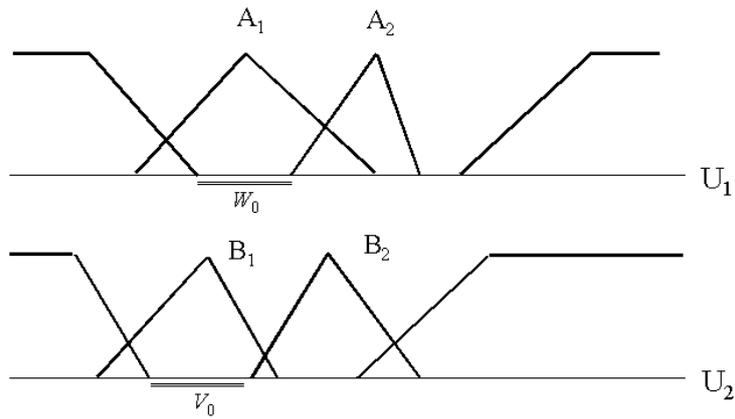


FIGURA 4.6 – Regiões não cobertas pelas regras da base.

A abordagem baseada em relações de similaridade segundo (Godo e Sandri, 2002a) considera um tipo de regra de inferência extrapolativa da forma:

$$\frac{\text{Se } X \text{ é } A \text{ então } Y \text{ é } B}{\text{Se } X \text{ é aproximadamente } \neg A \text{ então } Y \text{ é } B} \quad (4.10)$$

onde $\text{aproximadamente } \neg A = S \circ A$.

Esta regra é, de fato, um mecanismo de extrapolação, considerando-se que o limite de aplicabilidade da regra original é ampliado. Assim, pode-se observar que o novo conjunto de regras obtido é mais informativo, o que faz com que o problema de cobertura, do ponto de vista da perda mínima de informação, seja encontrar uma base de regras o mais próxima possível da base de regras que compõe o sistema original.

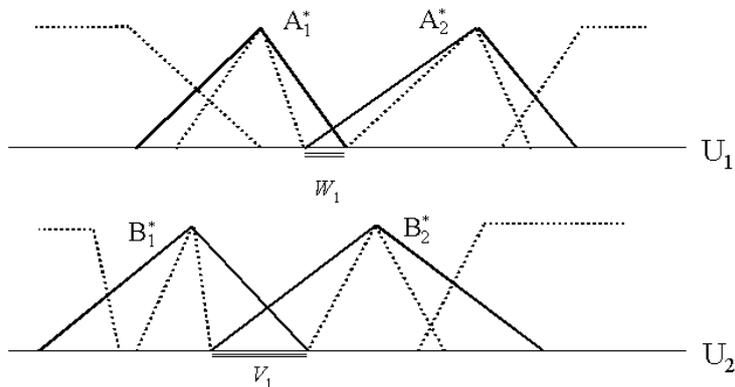


FIGURA 4.7 – Termos extrapolados, cobrindo as regiões W_1 e V_1 .

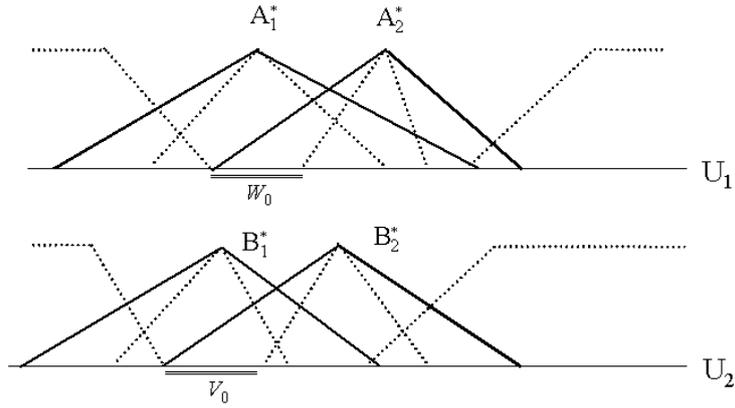


FIGURA 4.8 – Termos extrapolados, cobrindo as regiões W_0 e V_0 .

Portanto, é necessário verificar os possíveis problemas, ou seja, se a partição das variáveis de entrada envolvidas apresentam termos difusos que cobrem todo o espaço de entrada, e se a base de regras é tal que, para qualquer combinação da entrada, pelo menos uma regra é ativada. Determinada uma região problemática, um método é aplicado para encontrar as regras mais próximas às regiões não cobertas e a estas regras aplica-se uma relação de similaridade.

Considere como partições do espaço de entrada aquelas representadas na Figura 4.5, e $\Omega = (W_1, V_1)$ a região dos pontos não cobertos pelos termos A_1, A_2, B_1, B_2 , que definem as variáveis. Sejam S^1 e S^2 duas famílias de relações de similaridade em U_1 e U_2 respectivamente. Para cada regra R_j em K , define-se:

$$\lambda_j = \inf \{ \lambda | Su(S_\lambda^1 \circ A_j) \supseteq W_1 \}$$

$$\delta_j = \inf \{ \delta | Su(S_\delta^2 \circ B_j) \supseteq V_1 \}$$

Observa-se que, quanto maiores os valores de λ_j e δ_j , mais deve-se aumentar os termos difusos A_j e B_j para cobrir as regiões W_1 e V_1 , respectivamente.

Já para a segunda situação, considerando-se a relação de similaridade apresentada em 4.1.3, a abordagem consiste em substituir as regras R_1 e R_2 , definidas na seção anterior, pelas regras:

$$R_1^*: \text{ Se } X_1 \text{ é } A_1^* \text{ e } X_2 \text{ é } B_2^* \text{ então } Y \text{ é } C_1$$

$$R_2^*: \text{ Se } X_1 \text{ é } A_2^* \text{ e } X_2 \text{ é } B_1^* \text{ então } Y \text{ é } C_2$$

onde $A_i^* = S_{\lambda_0} \circ A_i$ e $B_i^* = S_{\delta_0} \circ A_i$, para $i = 1, 2$ com:

$$\lambda_0 = \inf \{ \lambda | S_\lambda \circ A_1 \supseteq W_0, S_\lambda \circ A_2 \supseteq W_0 \}$$

$$\delta_0 = \inf \{ \delta | S_\delta \circ B_1 \supseteq V_0, S_\delta \circ B_2 \supseteq V_0 \}$$

Assim, os termos difusos têm seus suportes modificados, de forma que as regiões W_0 e V_0 sejam cobertas. É possível verificar que $\lambda_0 = comprimento(W_0)$, $\delta_0 = comprimento(V_0)$

e que os termos produzidos são aqueles apresentados na Figura 4.8 (Godo e Sandri, 2002a).

Uma vez que a abordagem empregada para solucionar o problema da cobertura também se baseia em similaridade, estratégias global e local podem ser definidas na aplicação das relações, de forma análoga às estratégias definidas para o caso da consistência.

CAPÍTULO 5

DETERMINAÇÃO DE RELAÇÕES DE SIMILARIDADE POR PERDA MÍNIMA DE INFORMAÇÃO

O objetivo deste capítulo é apresentar a abordagem de perda mínima de informação na utilização de relações de similaridade para garantir 1-consistência (vide seção 4.1.2) em uma base de regras cuja inferência é feita utilizando-se implicações residuais ou o operador de Rescher-Gaines ((Drummond et al., 2002)). Uma condição suficiente para que uma base de regras com operadores residuais seja 1-consistente é que a base seja 1-consistente utilizando-se o operador de Rescher-Gaines. Por esta razão, este capítulo enfoca, particularmente, este operador.

Além da descrição dos procedimentos desenvolvidos empregando similaridade para garantir 1-consistência, este capítulo apresenta também uma aplicação da abordagem em controle difuso, trazendo os resultados obtidos e uma análise do comportamento dos controladores envolvidos nos experimentos.

Os testes foram realizados a partir de um sistema simples de controle difuso, disponível no ambiente ©Matlab, descrito na seção 3.6. Este ambiente de programação foi empregado para a implementação dos procedimentos envolvidos no projeto, com a utilização dos pacotes relacionados à lógica difusa e à simulação de sistemas de controle.

Os procedimentos desenvolvidos para solucionar o problema da consistência em sistemas implicativos com perda mínima de informação envolvem a verificação da 1-consistência da base de regras e o cálculo da menor relação de similaridade que torna uma base de regras 1-consistente. A próxima seção contém a descrição de como é feita a verificação de inconsistência e a seção seguinte apresenta o cálculo do parâmetro.

5.1 Verificação de inconsistências

Segundo (Dubois et al., 1996), quando todos os termos das variáveis (tanto de entrada quanto de saída) forem normalizados e convexos, a verificação de inconsistências em uma base de regras pode ser feita verificando-se a consistência entre cada par de regras. Portanto, para uma base composta por k regras, é necessária a verificação da consistência de $((k * (k + 1))/2) - k = k/2 * (k - 1)$ pares de regras. Se a base de regras representa todas as combinações de n variáveis de entrada com m termos cada, então $k = m^n$. Neste caso, portanto, o número de verificações é dado por $m^n/2 * (m^n - 1)$, ou seja, é da ordem de $O(m^{n^2}) = O(m^{2n})$. Por exemplo, para a Tabela 5.1 são necessárias 36 verificações.

Quando cada elemento da entrada pode estar contido no suporte de, no máximo, r termos da variável de entrada correspondente, o número de verificações pode ser reduzido. Neste caso, basta verificar-se a consistência nos subconjuntos de r^n regras contíguas da base. Quando $r = 2$ e $n = 2$, as regras que, necessariamente, devem ser verificadas entre si, formam arranjos tipo quadrado e tipo cubo quando $r = 2$ e $n = 3$. O número de conjuntos a serem verificados é de $(m - r + 1)^n$. Portanto, o número de verificações total, usando esta abordagem, é $(m - r + 1)^n * r^n$. Por exemplo, utilizando o valor típico $r = 2$, o número de verificações é $(m - 1)^n * 2^n$ é de m^n tanto para $r = 1$ quanto para $r = m$. Na Tabela 5.1, onde $r \leq 2$, são necessárias 16 verificações.

Se as verificações devem ser feitas repetidamente (por exemplo quando os termos são aprendidos) sobre uma estrutura fixa de regras e um mesmo valor de r , uma redução sobre o conjunto de regras por bloco pode ser obtida. Com efeito, considerando-se que os termos de saída são ordenados, para cada grupo basta armazenar uma única verificação que é tomada pelo primeiro e último termos envolvidos naquele grupo. Neste caso, o número máximo de verificações será de $(m - r + 1)^n$ e portanto, a complexidade não se altera. No entanto, no caso médio uma redução significativa pode ser obtida. Por exemplo, para a base da Tabela 5.1, com $r = 2$, o número de verificações se reduz a 4.

TABELA 5.1 – Base de regras para a variável *cold*.

e_flow/e_temp	<i>cold</i>	<i>good</i>	<i>hot</i>
<i>soft</i>	<i>os</i> (1)	<i>os</i> (4)	<i>of</i> (7)
<i>good</i>	<i>cs</i> (2)	<i>st</i> (5)	<i>os</i> (8)
<i>hard</i>	<i>cf</i> (3)	<i>cs</i> (6)	<i>cs</i> (9)

Um dos 4 “quadrados” a serem verificados na Tabela 5.1 é aquele formado pelas regras $\{R_1, R_2, R_4, R_5\}$. Neste conjunto de regras, detecta-se, por exemplo, uma inconsistência entre o par de regras $\{R_1, R_2\}$. Efetivamente, para a entrada $(-15, -0.2)$, correspondente a (e_temp, e_flow) , será disparada a regra R_1 , com grau de compatibilidade de 0.25, e a regra R_2 , com grau de 0.5. Utilizando o operador de Rescher-Gaines, estas regras levam às saídas $[open-slow]_{0.25} = [0.075, 0.525]$ e $[close-slow]_{0.5} = [-0.45, -0.15]$, cuja intersecção é vazia, ou seja, a base é inconsistente para esta entrada. Os outros pares de regras inconsistentes nesta base são $\{R_2, R_4\}$ (que levam à mesma inconsistência detectada para $\{R_1, R_2\}$), $\{R_3, R_5\}$, $\{R_5, R_7\}$, $\{R_6, R_8\}$ e $\{R_8, R_9\}$.

5.2 Cálculo do valor de λ

Verificada a área de conflito (que possivelmente gera inconsistência), aplica-se uma relação de similaridade para solucionar o problema. Este capítulo se restringe à utilização de uma família de relações de similaridade, apresentada na seção 4.1.3, a relação S_λ , que produz uma alteração simples em um conjunto difuso descrito por uma quádrupla $A = \langle a, b, c, d \rangle$, transformando-o em $S_\lambda \circ A = \langle a - \lambda, b, c, d + \lambda \rangle$. Além da simplicidade, outra vantagem desta relação é que ela não modifica o núcleo do conjunto difuso, ou seja, o intervalo $[b, c]$ não sofre alteração. A aplicação desta relação de similaridade ao conseqüente das regras, com um λ adequado, altera os termos difusos, fazendo com que haja intersecção entre eles e possibilitando que o mecanismo de inferência implicativo produza como saída um conjunto não vazio.

O cálculo do valor de λ , para garantir 1-consistência nos sistemas implicativos aqui considerados, é feito computando-se o suporte dos termos conseqüentes, B_1 e B_2 das regras ativadas ((Drummond et al., 2002)):

$$\lambda = \max\{E(Su(B_2)) - D(Su[B_1]_\delta), E(Su[B_2]_\delta) - D(Su(B_1))\} \quad (5.1)$$

onde $E([a, b]) = a$, $D([a, b]) = b$ e δ é o grau máximo de ativação na premissa das regras (vide seção 4.1.2).

Por exemplo, as inconsistências existentes na base de regras de ambas as variáveis de saída *cold* e *hot* desaparecem utilizando S_λ , com $\lambda = 0.3$, calculado por (5.1). Os termos de saída, após a aplicação de $S_{0.3}$, são apresentados na Figura 5.1 a). Pode-se observar que os conjuntos *open-slow* e *close-slow* passam a ser representados por $open-slow^{*0.3} = [-0.3, 0.3, 0.3, 0.9]$ e $close-slow^{*0.3} = [-0.9, -0.3, -0.3, 0.3]$. Assim, para a entrada $(-15, -2)$ são obtidas as saídas $[open-slow^{*0.3}]_{0.25} = [-0.15, 0.75]$ e $[close-slow^{*0.3}]_{0.5} = [-0.6, 0]$, com intersecção $[-0.15, 0]$.

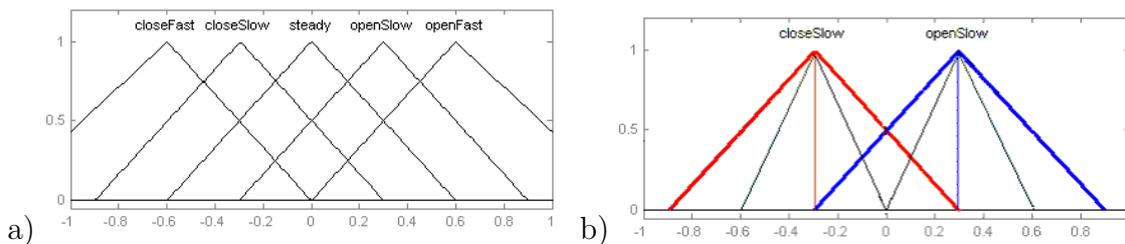


FIGURA 5.1 – a) Espaço de saída modificado utilizando $S_{0.3}$, b) detalhe.

5.3 Experimentos

Neste trabalho, foram desenvolvidos experimentos considerando o sistema *shower* onde os problemas de cobertura não são verificados.

O ©Matlab permite que diferentes mecanismos de inferência sejam implementados, modificando-se os métodos que implementam as operações de conjunção e disjunção de premissas, implicação, agregação e decodificação, que compõem o mecanismo. São empregados os seguintes operadores de implicação:

- Mamdani
- Rescher-Gaines
- Gödel

Na inferência de Mamdani a implicação é implementada pela função min e a agregação pela função max, constituindo um sistema conjuntivo. Nos outros dois mecanismos citados, a função que implementa os operadores (Rescher-Gaines e Gödel) é utilizada como método de implicação e a agregação é implementada pela função min. Estes são sistemas implicativos. Neste capítulo, a ênfase é na implicação de Rescher-Gaines, com os resultados sendo comparados àqueles obtidos para o operador de Mamdani.

O método de decodificação empregado foi o mesmo para todas as configurações do sistema, o centro de área (COA), onde o valor de saída é o centro de gravidade do conjunto difuso produzido na agregação.

Neste trabalho, foram empregadas as abordagens global, local e completamente local. Na abordagem global, a relação de similaridade é aplicada a todos os termos e a base de regras com os conjuntos modificados é utilizada para todas as entradas apresentadas ao sistema. A estratégia local aplica a relação de similaridade a todos os termos; porém, a nova base de regras é utilizada apenas para as entradas cujas regras ativadas se encontram em áreas de conflito. Para as demais entradas é ativada a base de regras original. A abordagem global se mostra vantajosa por ser uma aplicação direta, cuja programação é simples, além do tempo de execução ser curto. Por outro lado, as entradas que não ativam regras conflitantes, produzindo saídas perfeitamente possíveis, são afetadas de forma negativa pela imprecisão que a relação de similaridade aplicada pode gerar.

A terceira abordagem, chamada de completamente local, encontra uma relação de similaridade específica para cada entrada, aplicando-a somente aos termos envolvidos nas regras ativadas por aquela entrada em particular. Esta abordagem tem um custo

muito alto e pode ser inviável em algumas aplicações. Por ser de interesse reduzido, a estratégia completamente local não é abordada no restante deste trabalho.

5.4 Resultados

Os testes que empregam relações de similaridade na solução de problemas de inconsistência foram feitos utilizando-se o experimento *shower*, descrito na seção 3.6.

A Figura 5.2 mostra duas superfícies de controle para a variável *cold*¹. Na Figura 5.2 a) é ilustrada a superfície referente à abordagem tradicional de Mamdani e, na Figura 5.2 b), está a superfície obtida para o operador de Rescher-Gaines, utilizando os mesmos termos nas regras. Na Figura 5.2 b), muitas das saídas iguais a 0 correspondem a problemas de inconsistência na base de regras. Isto ocorre porque a entrada apresentada ativa regras cujos termos conseqüentes não têm intersecção, resultando em um conjunto vazio que produz o valor 0, quando decodificado.

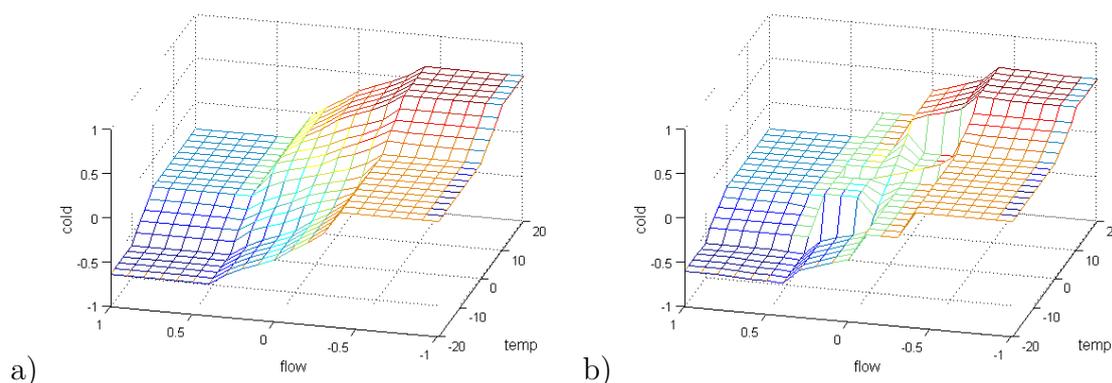


FIGURA 5.2 – Superfícies para a variável de saída *cold*: a) Mamdani, b) Rescher-Gaines com S_0 .

Aplicando a relação de similaridade adequada, superfícies de controle similares àquela obtida através da abordagem de Mamdani podem ser obtidas utilizando-se o operador de implicação de Rescher-Gaines, conforme Figura 5.3.

Para a família de relações de similaridade $S_{\lambda,1}$ apresentada em 4.1.3, o valor de λ igual a 0.3, calculado pela equação (5.1), é suficiente para gerar uma base de regras consistente.

A seguir, serão apresentados os gráficos resultantes da simulação feita no pacote *Simulink* do ©Matlab. A estrutura do sistema pode ser verificada na Figura 3.6, no capítulo 3. A

¹A título de simplificação são apresentadas apenas as superfícies de controle para a variável *cold*.

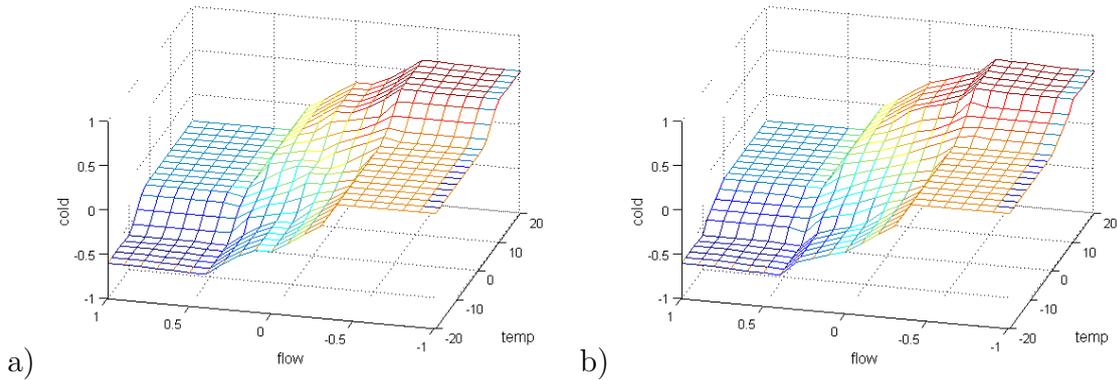


FIGURA 5.3 – Superfícies para a variável de saída *cold*: a) Rescher-Gaines com $S_{0,3}$ global, b) Rescher-Gaines com $S_{0,3}$ local.

Figura 5.4 apresenta os gráficos para as variáveis de entrada *temp* e *flow*, mediante um limiar (*set point*). As Figuras 5.5 e 5.6 apresentam a saída do controle difuso.

Nos itens a) e d) da Figura 5.4 e no item a) da Figura 5.5 são apresentados os gráficos obtidos para um sistema conjuntivo, segundo abordagem de Mamdani. Os gráficos para um sistema implicativo, empregando o operador de Rescher-Gaines são mostrados nos itens b) e e) da Figura 5.4 e b) da Figura 5.5. Pode-se observar que o sistema implicativo, utilizando os termos originais, por ser inconsistente, não funciona. Os itens c) e g), e d) e h) da Figura 5.4, e os itens a) e b), da Figura 5.6, trazem os gráficos relativos à aplicação de $S_{0,3}$, usando as abordagens global e local respectivamente.

Observa-se na Figura 5.6 que os resultados gerados, utilizando as abordagens global e local com o operador de Rescher-Gaines, são bem próximos daquele apresentado pelo método de Mamdani, sem muitas oscilações.

5.5 Desempenho dos controladores difusos

Para avaliar o desempenho do sistema *shower*, algumas funções de custo foram testadas. Foi, então, adotada a função:

$$f = soma1 + k * soma2, \quad (5.2)$$

onde *soma1* e *soma2* correspondem à soma dos erros absolutos referentes às variáveis *temp* e *flow*, respectivamente, considerando-se o limiar para cada uma das variáveis durante uma simulação padrão. A constante *k* serve para ponderar dois termos, e foi tomada como $k = l_1/l_2$, onde l_1 e l_2 correspondem à amplitude máxima dos limiares na simulação. No caso, $l_1 = 8$ e $l_2 = 0.4$ e, conseqüentemente, $f = soma1 + 20 * soma2$.

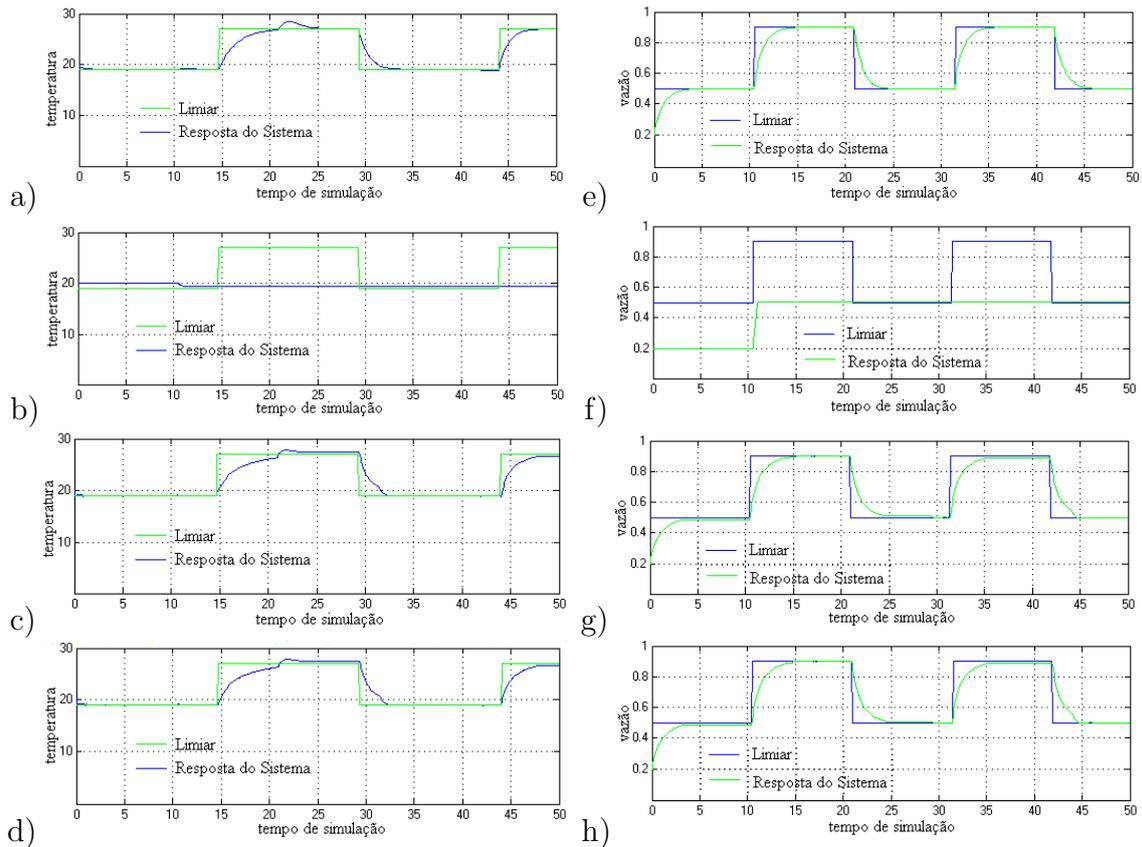


FIGURA 5.4 – Variável *temp*: a) Mamdani, b) Rescher-Gaines com S_0 , c) Rescher-Gaines com $S_{0.3}$ global, d) Rescher-Gaines com $S_{0.3}$ local.

TABELA 5.2 – Valores da função de avaliação

	Mamdani	RG com $S_{.3}$ local	RG com $S_{.3}$ global
f	650.11	814.68	827.96

A função empregada possibilita a comparação entre dois sistemas de diferentes naturezas, implicativa e conjuntiva. Os valores da função podem ser observados na Tabela 5.2.

Os valores obtidos são para o sistema *shower*, empregando o mecanismo de inferência de Mamdani, que apresenta melhor desempenho, e empregando o mecanismo de inferência de Rescher-Gaines, com uma relação de similaridade $\lambda = 0.3$, para as abordagens global e local.

5.6 Emprego do operador de Gödel

A Figura 5.7 apresenta as superfícies de controle obtidas empregando-se o operador de implicação de Gödel, considerando a estratégia local com $\lambda = 0.3$ e $\lambda = 0.1$.

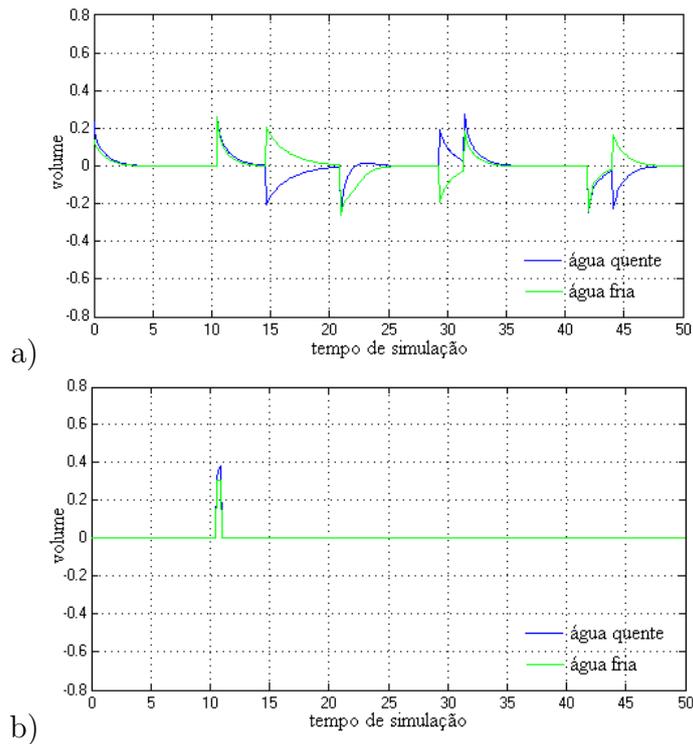


FIGURA 5.5 – Saída do controlador: a) Mamdani, b) Rescher-Gaines com S_0 .

Como as R -implicações produzem resultados maiores do que aqueles produzidos pelo operador de Rescher-Gaines, é possível obter uma base de regras consistente com um valor de λ menor que o obtido para Rescher-Gaines. Em particular, para o operador de Gödel, o valor de λ que resolve todas as inconsistências deste sistema é qualquer valor maior do que 0.

Os gráficos de simulação obtidos são apresentados nas Figuras 5.8, 5.9 e 5.10. Vê-se na Figura 5.10 que o emprego da implicação de Gödel não apresentou uma saída satisfatória, com muitas oscilações tanto para $\lambda = 0.3$ como para $\lambda = 0.1$. Aqui são apresentados os resultados obtidos apenas para a abordagem local, tendo a abordagem global apresentado desempenho inferior.

Nota-se que as oscilações na variável $temp$ surgem quando o limiar (*set point*) na variável $flow$ decresce. Já o acréscimo no limiar da variável $flow$, aparentemente, não provoca alterações. As oscilações na variável $flow$ parecem ser independentes da variação na variável $temp$.

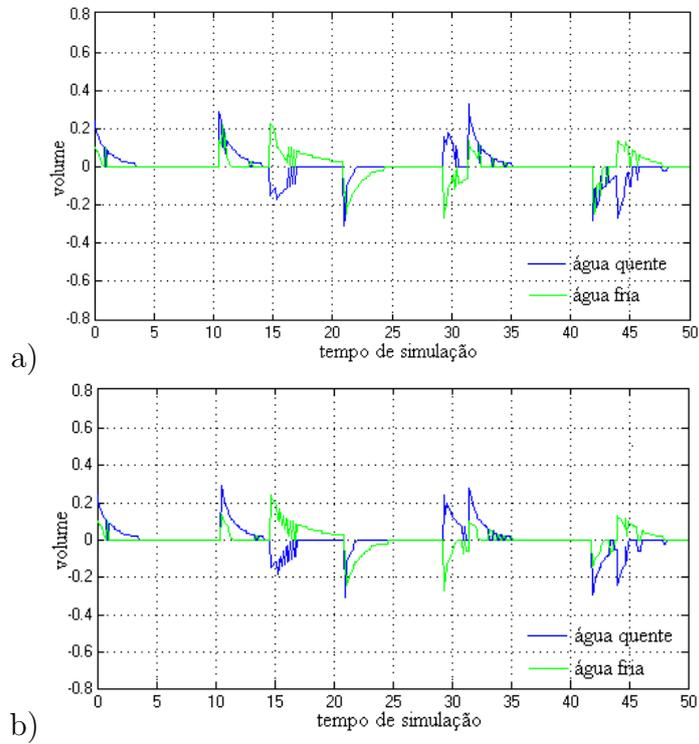


FIGURA 5.6 – Saída do controlador: a) Rescher-Gaines com $S_{0.3}$ global, b) Rescher-Gaines com $S_{0.3}$ local.

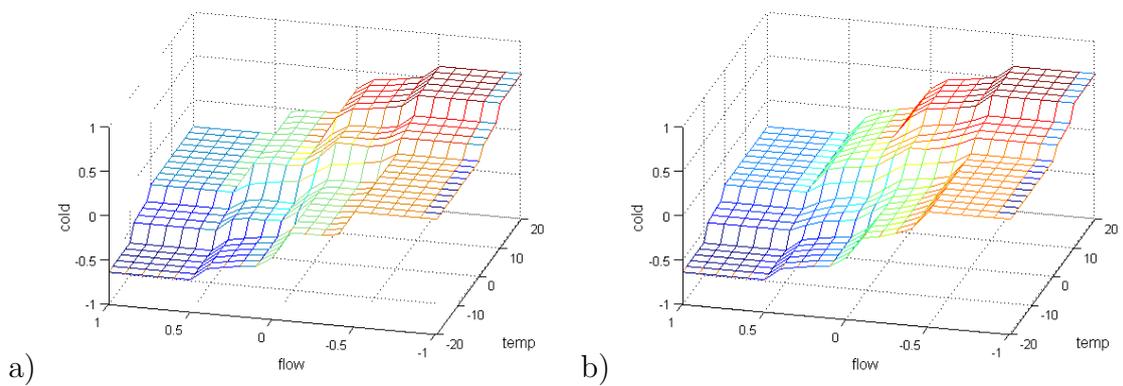


FIGURA 5.7 – Superfícies para a variável de saída *cold*: a) Gödel com $S_{0.1}$ local, b) Gödel com $S_{0.3}$ local.

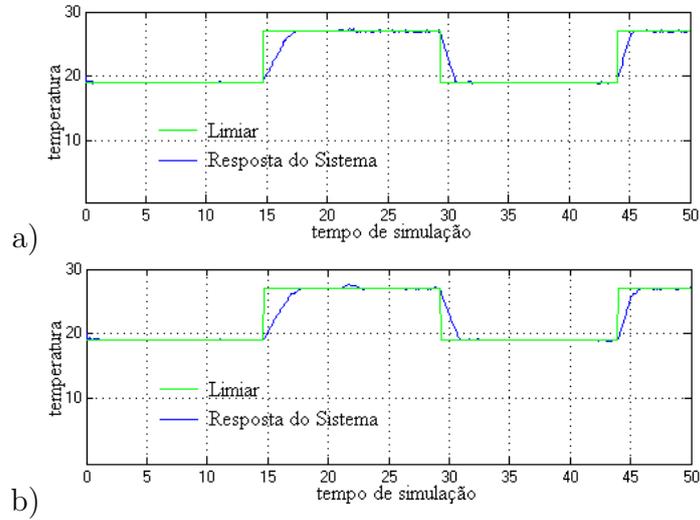


FIGURA 5.8 – Variável *temp*: a) Gödel com $S_{0,1}$ local, b) Gödel com $S_{0,3}$ local.

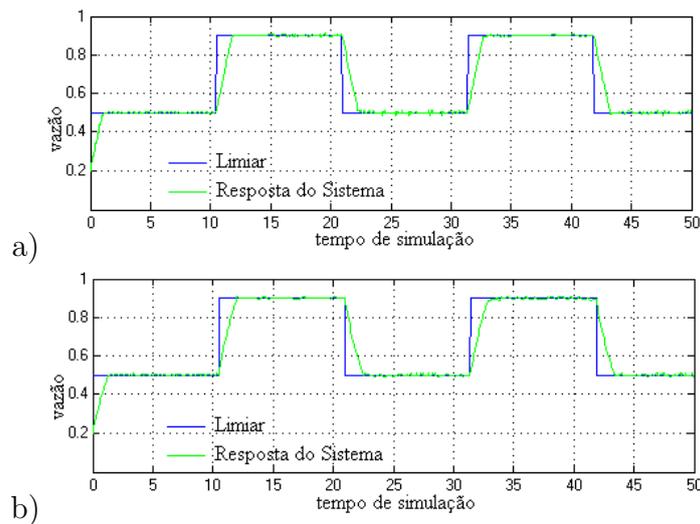


FIGURA 5.9 – Variável *flow*: a) Gödel com $S_{0,1}$ local, b) Gödel com $S_{0,3}$ local.

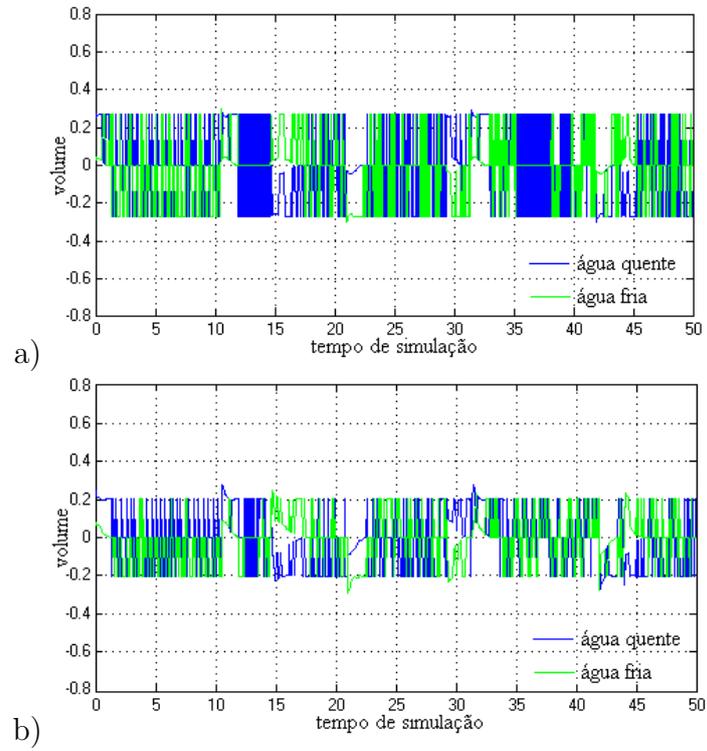


FIGURA 5.10 – Saída do controlador: a) Gödel com $S_{0.1}$ local, b) Gödel com $S_{0.3}$ local.

CAPÍTULO 6

DETERMINAÇÃO DE RELAÇÕES DE SIMILARIDADE POR APRENDIZADO

Embora o critério de perda mínima de informação garanta a consistência de um sistema implicativo, os resultados apresentados na Tabela 5.2 demonstraram claramente que esta abordagem não garante, necessariamente, o melhor desempenho de um dado sistema. Na busca por melhores resultados, é proposta uma nova abordagem que se baseia em aprendizado.

O objetivo deste capítulo é apresentar uma abordagem que permite determinar as relações de similaridade por aprendizado para tratar problemas de α -consistência, para $\alpha > 0$, e falta de cobertura verificados numa base de regras de um sistema difuso.

Os experimentos de aprendizagem consideram aplicações cuja base de regras é conhecida. Como, neste trabalho, os termos são representados por funções triangulares e trapezoidais, é necessário aprender somente o suporte e o núcleo dos termos difusos que compõem as variáveis, além dos parâmetros relacionados às relações de similaridade envolvidas. São empregados os sistemas de Gödel, Goguen e Mamdani.

Como técnica de aprendizado, foram empregados os algoritmos genéticos, visando encontrar os melhores parâmetros para uma determinada aplicação. Uma breve revisão da literatura sobre os algoritmos genéticos é apresentada na seção seguinte. O capítulo abrange, ainda, como é feito o aprendizado dos termos difusos, quais as características do AG implementado e as melhores soluções encontradas para os experimentos realizados.

6.1 Algoritmos genéticos

Uma maneira de se aprender parâmetros de sistemas difusos consiste no uso de algoritmos genéticos (AG)(Pedrycz, 1998). Os algoritmos genéticos constituem um método de busca adaptativa em que se busca a solução ótima para um dado problema. São baseados na analogia entre otimização e evolução natural das espécies, combinando os conceitos de adaptação seletiva e sobrevivência dos indivíduos mais capazes (Goldberg, 1989).

Nestes algoritmos, define-se uma população de indivíduos, onde cada indivíduo representa uma possível solução para um dado problema. Cada indivíduo é reflexo de uma estrutura de dados simples, usualmente denominada cromossoma, que sofre transformações através da aplicação das operações de seleção, cruzamento e mutação. A qualidade de cada indivíduo é avaliada através de uma medida que reflete sua adequação ao ambiente. A

natureza favorece a reprodução daqueles indivíduos considerados mais aptos dentro da população. Existe, portanto, uma competição pela sobrevivência entre os indivíduos, ou seja, seleção natural.

Os estágios de um algoritmo genético podem ser descritos como se segue. Inicialmente, cria-se uma população de indivíduos, através de um processo aleatório. Os indivíduos são avaliados através da função objetivo (“fitness”), que verifica o quanto a solução representada pelo cromossoma da população se aproxima daquela considerada ideal. Os indivíduos sofrem uma seleção, sendo que os indivíduos melhor avaliados têm maior probabilidade de serem selecionados para reprodução. A nova geração é obtida através da aplicação dos operadores de cruzamento e mutação. O cruzamento toma dois indivíduos (pais) e combina-os aleatoriamente gerando filhos que apresentam combinações das características dos pais. Já a mutação introduz uma pequena modificação em um cromossoma selecionado e é aplicada de forma aleatória, mas de modo a não descaracterizar a população. Desta maneira, evita-se perder as características herdadas dos melhores indivíduos, mas permite-se maior variabilidade na população (e, em consequência, uma menor probabilidade de ocorrência de mínimos locais).

Todo esse processo é repetido a cada geração, executando-se as operações genéticas no sentido de aprimorar a população, até que a solução ótima seja encontrada.

6.2 Aprendizado de termos difusos

As famílias de relações de similaridade empregadas nesta abordagem são aquelas apresentadas na seção 4.1.3. A seguir será descrito como as relações de similaridade podem tratar problemas de inconsistência e de falta de cobertura em sistemas de regras difusas através de aprendizado, supondo-se que a base de regras é previamente conhecida. Supondo-se que as regras apresentem n_1 variáveis na entrada e n_2 na saída, num total de $n = n_1 + n_2$ variáveis, algumas situações são possíveis:

- a) A base de regras é completa (vide capítulo 4). Neste caso, o aprendizado de relações de similaridade serviria somente para se tentar melhorar o desempenho do sistema. Entretanto, o custo de aprendizado dos parâmetros das relações pode ser alto em relação ao resultado obtido, mesmo levando-se em conta que a tarefa de aprendizado seria realizada uma única vez.
- b) A base de regras cobre todo o espaço de entrada, mas é inconsistente. Neste caso, é possível empregar $S_{\lambda_i,1}$, S_{0,γ_i} ou S_{λ_i,γ_i} para tratar o problema. Portanto, são aprendidos 1 ou 2 parâmetros para cada variável de saída, num total de n_2 ou $2 * n_2$ parâmetros.

- c) A base de regras é consistente, mas apresenta problemas de falta de cobertura. Este caso é análogo ao caso anterior, porém envolve as variáveis de entrada e não as variáveis de saída, ou seja, deve-se aprender n_1 ou $2 * n_1$ parâmetros.
- d) A base de regras apresenta tanto problemas de consistência quanto problemas de cobertura. Neste caso, o número de parâmetros envolvidos é igual a n ou $2*n$. Particularmente, considera-se o caso de termos representados por pontos, ou seja, a entrada e a saída são valores precisos. Um tratamento para esta configuração seria adotar os valores precisos como núcleo dos termos difusos. Outra possibilidade é a de empregar a relação S_{0,γ_i} para modificar os núcleos, aprendendo, assim, 1 parâmetro extra por variável.
- e) Apenas o número de termos difusos para cada uma das variáveis é conhecido, ou seja, a base é totalmente incompleta. Neste caso, é necessário aprender tanto o núcleo quanto o suporte dos termos. Se se optar por aprender núcleos precisos, é necessário aprender-se um ponto para cada núcleo e 1 ou 2 parâmetros da relação de similaridade envolvida, que estabelecerão os suportes.

As situações (a) e (c) podem ocorrer tanto em sistemas conjuntivos quanto em sistemas implicativos, e (b) apenas em sistemas implicativos, uma vez que os sistemas conjuntivos nunca se apresentam inconsistentes. As situações (d) e (e) podem também ocorrer para os dois tipos de sistemas, levando-se em consideração que, para os sistemas implicativos, é necessário fazer verificações relacionadas aos problemas de cobertura e consistência, enquanto que, nos sistemas conjuntivos, apenas a cobertura deve ser garantida.

A abordagem global é mais interessante quando os termos são aprendidos, uma vez que o custo da abordagem local é alto, o que, aliado ao alto custo dos algoritmos de aprendizado em si, pode tornar inviável a aprendizagem.

6.3 Características do AG implementado

Neste trabalho, emprega-se um algoritmo genético (AG) simples com cruzamento e mutação para as tarefas de aprendizado. A base de regras que compõe o sistema é sempre conhecida e os termos difusos são sempre descritos por funções triangulares ou trapezoidais. Dessa forma, cada cromossoma codifica os parâmetros tais como descritos na seção 6.2. Cada parâmetro codificado é chamado de gene (Lacerda e Carvalho, 1999).

Para reduzir o número de parâmetros codificados pelo cromossoma, considera-se que a partição dos termos difusos é simétrica e apresenta o *zero* como termo central, onde

$Nu(zero) = [0, 0]$. Desta forma, para aprender um número ímpar n de termos, é preciso codificar somente $(n - 1)/2$ genes. A Figura 6.1 traz uma partição simétrica de 3 termos: T_a , T_b e T_c (note que os termos T_a e T_c , em si, não são simétricos).

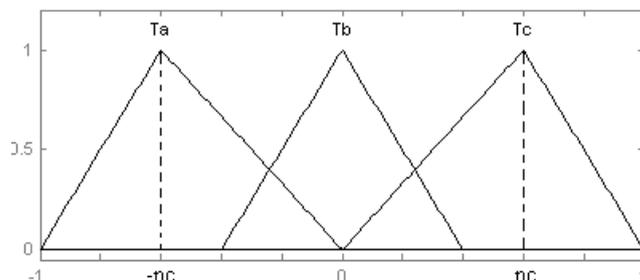


FIGURA 6.1 – Partição simétrica.

A população inicial é gerada aleatoriamente; porém, ao longo da evolução, são considerados apenas os indivíduos válidos, ou seja aqueles que formam bases completas. Para tanto, cada novo indivíduo gerado é testado e apenas aqueles que obedecem às restrições impostas são mantidos na população; os demais são eliminados.

No caso do aprendizado dos parâmetros para os sistemas implicativos, o procedimento que identifica indivíduos válidos na população verifica problemas de inconsistência e de falta de cobertura. Para os sistemas conjuntivos, apenas a cobertura é verificada. Como visto na seção 4.2, a falta de cobertura pode acontecer pela presença de *buracos* no espaço de saída, ou configuração inadequada de base de regras. O segundo problema não é tratado neste trabalho, visto que as bases aqui empregadas não apresentam problemas estruturais, ou seja, para qualquer combinação da entrada vai ser disparada pelo menos uma regra da base de regras, uma vez que as regras da base correspondem à combinação de todos os 3 termos difusos que definem as 2 variáveis de entrada.

Para garantir que a população seja formada apenas por indivíduos que cubram todo o espaço de entrada, o procedimento busca pelos *buracos*. São computados os valores dos suportes esquerdo e direito de cada termo que define as variáveis de entrada, considerando que o conjunto de termos difusos é dado por $\{T_1, \dots, T_n\}$, onde T_1, \dots, T_n são definidos no universo de discurso $[E_{lim}, D_{lim}]$ e são consecutivos (vide seção 3.2). As seguintes restrições são definidas:

- $E(Su(T_1)) \leq E_{lim}$
- $D(Su(T_n)) \geq D_{lim}$

- $D(Su(T_n)) \geq E(Su(T_{n+1}))$

Com relação à consistência, nenhuma verificação é feita nos experimentos que envolvem os sistemas de Mamdani, por serem desnecessárias (vide seção 4.1.1). Para os experimentos com os sistemas de Gödel e Goguen é feita a verificação de α -consistência, $\alpha > 0$ (vide seção 4.1.2).

Para o sistema *shower*, considerado neste trabalho, foi adotada a função de avaliação f , definida na seção 5.5, baseada na soma dos erros absolutos gerados mediante um limiar, durante a simulação do sistema.

O critério de parada utilizado é o número de gerações considerado para evolução de uma população de indivíduos válidos. Outros critérios de parada podem ser empregados, como por exemplo, estabelecer um erro desejado e parar a execução quando este valor for atingido. Porém, neste caso, a melhor solução não é conhecida não sendo trivial definir o erro desejado.

Para que seja possível explorar o espaço de busca, são empregados os operadores genéticos de cruzamento e mutação. Estes operadores devem ser aplicados de maneira que os indivíduos mais aptos têm maiores chances de evoluírem, passando suas características para as gerações seguintes. São selecionados dois indivíduos, o primeiro de forma aleatória e o segundo utilizando uma seleção proporcional, o método da roleta (Goldberg, 1989).

A operação de cruzamento é aplicada ao par de cromossomas selecionado, m e p , gerando apenas um descendente d . Foi implementado um cruzamento aritmético, descrito em (Michalewicz, 1996), onde cada gene d_i do descendente é obtido segundo a equação:

$$d_i = m_i + \beta(p_i - m_i) \quad (6.1)$$

com $\beta \in [0, 1]$ obtido aleatoriamente.

Após o cruzamento, é aplicado um operador de mutação sobre o descendente, com probabilidade p_{mut} igual a 5%. Na mutação, um gene é escolhido ao acaso e alterado para um valor aleatório dentro do domínio da variável, com distribuição uniforme.

6.4 Experimentos

Os experimentos descritos aqui foram realizados utilizando-se o sistema *shower* (vide seção 3.6), considerando-se tanto os sistemas implicativos quanto os sistemas conjuntivos e considerando-se a situação e) descrita na seção 6.2. Ou seja, são aprendidos valores

precisos para os núcleos na entrada e na saída, sendo dados apenas o número de termos difusos que definem cada uma das variáveis envolvidas.

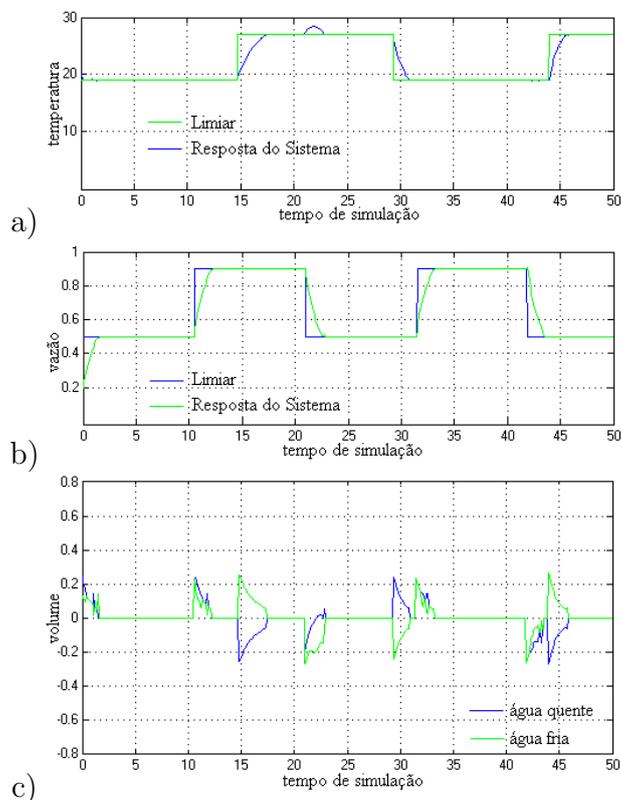


FIGURA 6.2 – Gödel: a) variável *temp*, b) variável *flow*, c) saída do controle.

O sistema *shower* apresenta duas variáveis de entrada, cada qual definida por 3 termos difusos consecutivos, Ta , Tb e Tc , conforme Figura 6.1¹. Para cada uma das variáveis, um único parâmetro é aprendido pelo algoritmo genético, correspondendo ao núcleo nc com os núcleos de Tb e Ta sendo dados, respectivamente, por 0 e $-nc$. O indivíduo é uma estrutura composta por 8 genes: um ponto para o núcleo da variável e_temp (p_t), valor de λ para a variável e_temp (λ_t), um ponto para o núcleo da variável e_flow (p_f), valor de λ para a variável e_flow (λ_f), dois pontos para os núcleos das variáveis de saída ($p1_s, p2_s$), e λ e γ para as variáveis de saída.

Nos experimentos descritos a seguir, para cada um dos mecanismos de inferência considerados no aprendizado, é apresentado o melhor indivíduo encontrado pelo AG, sendo tabelados os valores dos parâmetros obtidos para cada gene, bem como o valor de

¹Os experimentos descritos nesta seção envolvem termos difusos descritos por funções triangulares. Foram realizados testes envolvendo termos difusos descritos por funções trapezoidais, porém estes não apresentaram melhores resultados.

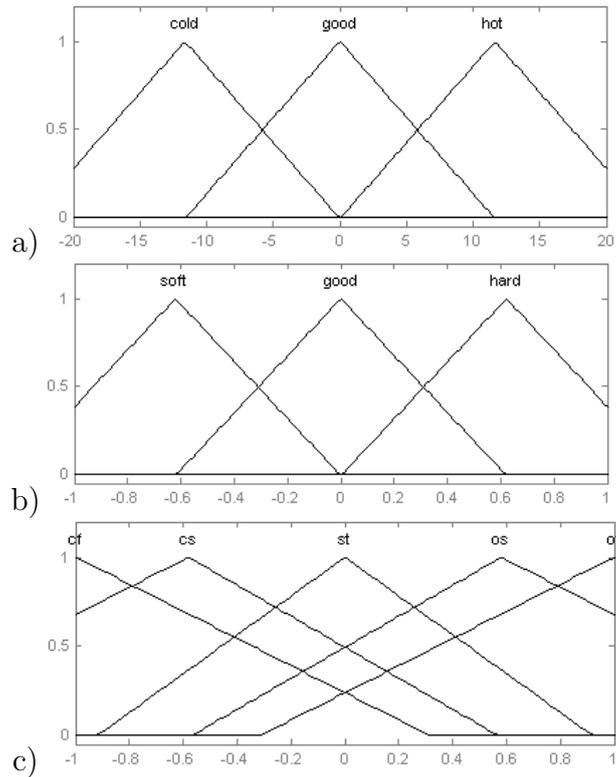


FIGURA 6.3 – Gödel: termos de entrada: a) variável e_temp , b) variável e_flow , c) termos que definem as variáveis de saída.

TABELA 6.1 – Melhores indivíduos

	p_t	λ_t	p_f	λ_f	$p1_s$	$p2_s$	λ_s	γ_s	f
Mamdani	11.3326	12.3733	0.5461	0.6723	0.5680	1.0000	0.1385	0.7927	320.16
Gödel	11.6397	11.5773	0.6209	0.6171	0.5805	1.0000	0.9180	0.6046	441.69
Goguen	13.6026	13.4912	0.5900	0.6691	0.4926	1.0000	0.6439	0.4820	591.12

aptidão do indivíduo. Além disso, são apresentados os termos difusos obtidos (Figuras 6.3, 6.5 e 6.7) e os gráficos resultantes da simulação do sistema *shower* (Figuras 6.2, 6.4 e 6.6), para as variáveis de entrada, $temp$ e $flow$, e para a saída do controle difuso.

Observa-se, na Tabela 6.1, que o desempenho dos melhores indivíduos aprendidos é superior àquele apresentado pelo sistema de Mamdani original (vide Tabela 5.2). Além disso, pode-se verificar que o sistema de Mamdani mostrou-se melhor do que os sistemas empregando os operadores de Gödel e Goguen, em relação à função de avaliação. No entanto, é importante notar que os desempenhos dos sistemas em si não apresentam grandes diferenças, como pode ser constatado nos gráficos de simulação, nas Figuras 6.2, 6.4 e 6.6.

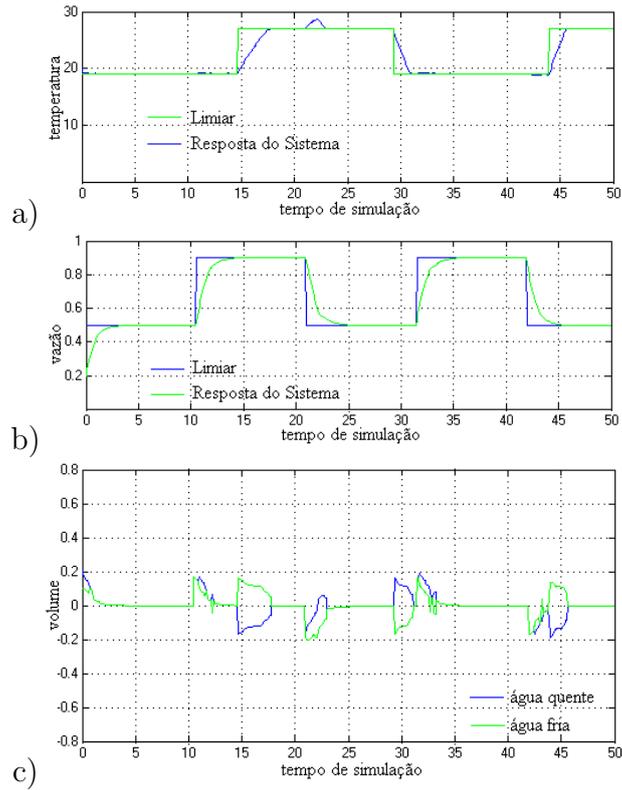


FIGURA 6.4 – Goguen: a) variável *temp*, b) variável *flow*, c) saída do controle.

Pode-se observar ainda que a configuração dos termos de saída para o melhor indivíduo encontrado para o sistema de Mamdani (Figura 6.7 c)) seria totalmente inconsistente em um sistema implicativo.

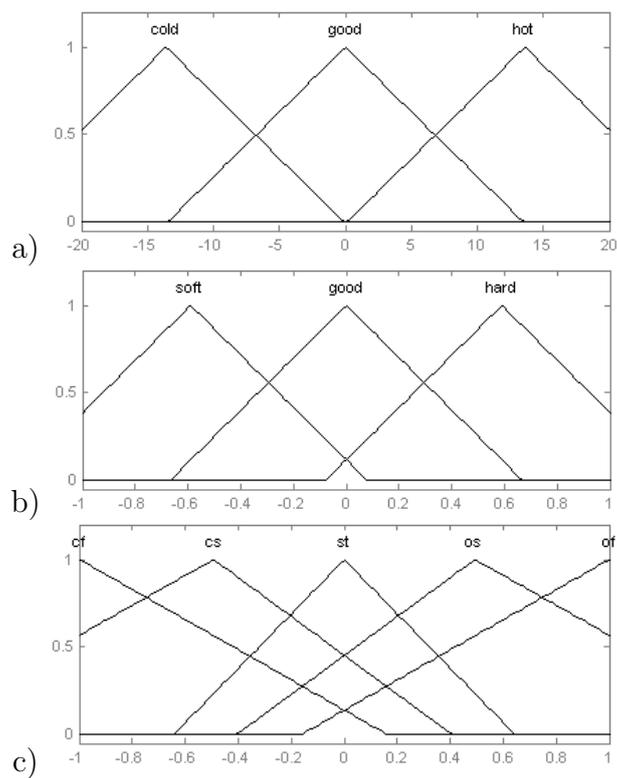


FIGURA 6.5 – Goguen: termos de entrada: a) variável e_temp , b) variável e_flow , c) termos que definem as variáveis de saída.

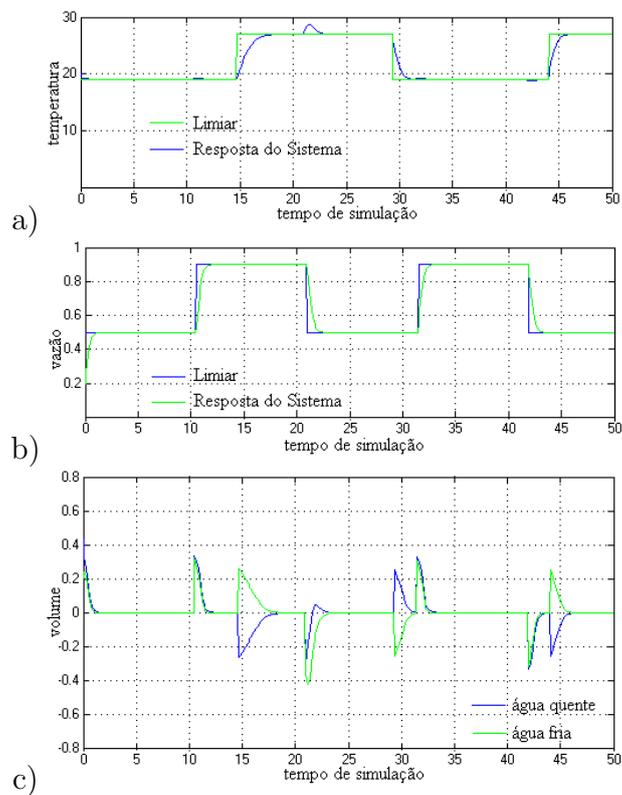


FIGURA 6.6 – Mamdani: a) variável $temp$, b) variável $flow$, c) saída do controle.

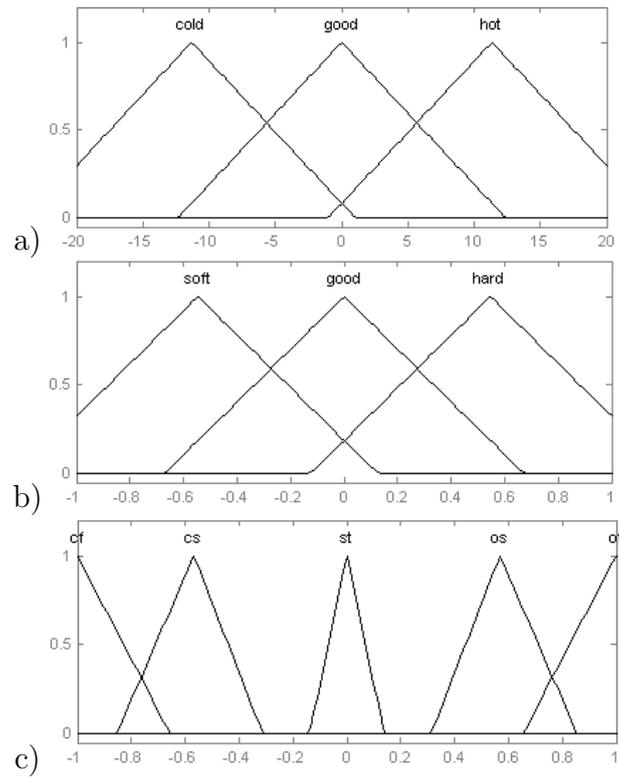


FIGURA 6.7 – Mamdani: termos de entrada: a) variável e_temp , b) variável e_flow , c) termos que definem as variáveis de saída.

CAPÍTULO 7

CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, foram utilizadas relações de similaridade para tratar consistência e cobertura em sistemas de regras difusas. Na solução dos problemas de inconsistência, os termos conseqüentes das regras são substituídos por outros mais imprecisos, obtidos através de uma relação de similaridade adequada. Para superar os problemas de falta de cobertura, modifica-se a premissa das regras envolvidas, tornando os conjuntos difusos mais imprecisos. Neste contexto, foram investigadas uma abordagem visando perda mínima de informação em sistemas com bases de regras pré-existentes e uma abordagem por aprendizado para um conjunto maior de aplicações. A abordagem por perda mínima de informação foi aplicada para tratar 1-consistência em sistemas implicativos que empregam os operadores residuais ou o operador de Rescher-Gaines. A abordagem por aprendizado foi aplicada tanto a sistemas conjuntivos quanto a sistemas de regras que utilizam operadores residuais de Gödel e Goguen, verificando-se, nestes últimos, a α -consistência, $\alpha > 0$.

Os procedimentos referentes ao projeto foram implementados através do ambiente ©MatLab, com a utilização dos pacotes relacionados ao problema, incluindo o sistema de controle utilizado para realizar os testes. A bateria de testes realizada empregando diferentes configurações foi a base da análise de desempenho dos controladores difusos tradicionais (que são baseados em sistemas conjuntivos) e dos sistemas implicativos.

O sistema de controle difuso *shower*, disponível no ambiente de programação ©Matlab, foi utilizado como exemplo. Este sistema emprega o controlador de Mamdani e os resultados obtidos na simulação foram utilizados para comparação de desempenho. A medida de comparação de desempenho considera o erro obtido em um determinado tempo de simulação.

Na abordagem que determina relações de similaridade por perda mínima de informação, são empregadas as estratégias global e local, para os sistemas implicativos de Rescher-Gaines e Gödel. Os desempenhos do sistema de Rescher-Gaines empregando as estratégias global e local sobre um conjunto de termos pré-existentes não apresentam diferenças significativas entre si. Por outro lado, a abordagem global mostrou-se melhor no que diz respeito ao custo computacional. Comparando-se ao sistema de Mamdani, para o qual os termos foram originalmente criados, este último apresentou melhor desempenho.

Como alternativa para tentar melhorar o desempenho dos sistemas, ao mesmo tempo assegurando a consistência e cobertura da base, foi empregado um algoritmo genético para encontrar os melhores parâmetros dos termos difusos e das relações de similaridade. Para esta abordagem, foram considerados os sistemas implicativos de Gödel e Goguen, que obtiveram resultados comparáveis aos obtidos pelo controlador de Mamdani. Uma desvantagem observada na técnica que envolve o aprendizado dos termos e das relações de similaridade foi o tempo de processamento necessário para obtenção de uma solução satisfatória.

A configuração que se mostrou melhor com relação aos resultados obtidos é aquela em que o algoritmo genético encontra os melhores termos difusos, tanto para as variáveis de entrada quanto para as variáveis de saída do sistema. Porém, este experimento requer um número maior de parâmetros a serem codificados pelo algoritmo genético, consumindo maior tempo de processamento.

É importante ressaltar que os experimentos levados a cabo neste trabalho foram baseados apenas em um sistema específico de controle, visando tão somente validar e ilustrar os resultados formais. Espera-se, no entanto, que os formalismos descritos aqui sejam igualmente bem-sucedidos quando aplicados a sistemas mais complexos.

Da mesma maneira, o algoritmo genético utilizado aqui pode ser refinado em busca de resultados melhores e mais adequados às aplicações. Além disso, outras técnicas de aprendizado podem ser igualmente utilizadas.

A maior contribuição deste trabalho é a viabilização do uso de operadores de implicação propriamente ditos em aplicações de sistemas difusos. Isto é interessante, pois estas implicações estão intuitivamente ligadas à noção de inferência, por analogia com a lógica clássica, mas não tem merecido atenção na literatura mais voltada ao desenvolvimento de aplicações.

Uma das razões para isso é a tendência de se utilizar as chamadas partições de Ruspini para sintonizar os termos difusos nas regras. Estas partições são muito úteis em sistemas conjuntivos mas, muito freqüentemente, geram inconsistências nos sistemas implicativos.

Este trabalho tem, pois, também como contribuição a proposta de modelo alternativo às partições de Ruspini, que pode ser empregado vantajosamente não só na criação de sistemas implicativos mas, também, de sistemas conjuntivos.

O uso de relações de similaridade também é potencialmente interessante em aplicações de raciocínio baseado em casos (Aamodt e Plaza, 1994). Nestes sistemas, armazena-se

uma grande quantidade de casos formados por pares (entrada, saída observada), onde a entrada é usualmente um vetor de grande dimensionalidade. Dado um novo caso c , para o qual não se conhece a saída, procura-se encontrar na base os casos cujas entradas mais se assemelham à entrada do caso c , para então, de alguma maneira, agregar a saída dos casos correlatos e assim fornecer uma saída para o caso c .

As relações de similaridade podem ser utilizadas para tornar mais imprecisos os valores de entrada do caso c e prover, com isso, uma medida de distância natural de c aos casos contidos na base. Também em sistemas que sejam criados desta maneira, existirá a necessidade de se tratar a consistência/cobertura na agregação das saídas dos casos similares a c . No entanto, o formalismo teórico para este tratamento será basicamente o mesmo que o apresentado aqui. A única diferença é que, nesses sistemas, será necessário escolher-se o número de casos ou um limiar de compatibilidade, para delimitar um conjunto de casos a serem agregados, já que não existirá uma estrutura de regras pré-estabelecida como nos sistemas tratados neste trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

Aamodt, A.; Plaza, E. Cased-based reasoning: foundational issues, methodological variations, and system approaches. **AI Communications**, v. 7, n. 1, p. 39–59, 1994. [74](#)

Bouchon-Meunier, B.; Dubois, D.; Godo, L.; Prade, H. Fuzzy sets and possibility theory in approximate and plausible reasoning. **Fuzzy Sets in Approximate reasoning and Information Systems**, v. 5, p. 15–190, 1999. [27](#)

Bouchon-Meunier, B.; Dubois, D.; Marsala, C.; Prade, H.; Ughetto, L. A comparative view of interpolation methods between sparse fuzzy rules. In: Joint IFSA World Congress and NAFIPS International Conference(2001), 9, Vancouver (Canada). **Proceedings**. Vancouver: IFSA/NAFIPS, 2001. p. 2499–2504. [47](#)

Driankov, D.; Hellendoorn, H.; Reinfrank, M. **An introduction to fuzzy control**. Berlin: Springer-Verlag, 1996. 350 p. [19](#), [20](#), [25](#), [31](#), [34](#), [35](#), [36](#)

Drummond, I.; Godo, L.; Sandri, S. Restoring consistency in systems of fuzzy gradual rules using similarity relations. In: Brazilian Symposium on Artificial intelligence(SBIA2002), 16, Recife(Brasil), 2002. **Advances in Artificial Intelligence**. Berlin: Springer, 2002. v. 2507, p. 386–396. [41](#), [42](#), [43](#), [51](#), [53](#)

———. Learning fuzzy systems with similarity relations. In: Fuzzy Systems Association World Congress(IFSA2003), Istanbul(Turkey), 2003. **Proceedings**. Berlin: Springer, 2003. Submitted. [41](#), [43](#)

Dubois, D.; Esteva, F.; Garcia, P.; Godo, L.; Mantaras, R. L.; Prade, H. Fuzzy set modelling in case-based reasoning. **International Journal of Intelligent Systems**, v. 13, n. 4, p. 345–373, 1998. [27](#), [28](#), [29](#)

Dubois, D.; Prade, H. Gradual inference rules in approximate reasoning. **Information Sciences**, v. 61, n. 1, p. 103–122, 1992. [32](#)

———. What are fuzzy rules and how to use them. **Fuzzy Sets and Systems**, v. 84, n. 2, p. 169–185, 1996. [19](#), [31](#), [32](#)

Dubois, D.; Prade, H.; Ughetto, L. Coherence of fuzzy knowledge bases. In: International IEEE Conference on Fuzzy Systems(FuzzIEEE'96), 5, New Orleans (EUA), 1996. **Proceedings**. EUA: IEEEPress, 1996. v. 1, p. 1858–1864. [20](#), [41](#), [51](#)

———. Checking the coherence and redundancy of fuzzy knowledge bases. **IEEE Transactions on Fuzzy Systems**, v. 5, n. 5, p. 398–717, 1997. [41](#)

Godo, L.; Sandri, S. Dealing with covering problems in fuzzy rule systems by similarity-based extrapolation. In: International IEEE Conference on Fuzzy Systems (FuzzIEEE'02), 11, Honolulu(EUA), 2002. **Proceedings**. Honolulu: IEEE Press, 2002a. v. 1, p. 723–728. [20](#), [47](#), [48](#), [50](#)

———. A similarity-based approach to deal with inconsistency in systems of fuzzy gradual rules. In: Information Processing and Management of Uncertainty in knowledge-Based Systems (IPMU'2002), 9, Annecy (France), 2002. **Proceedings**. Berlin: Springer, 2002b. v. 3, p. 1655–1662. [19](#), [20](#), [39](#), [40](#), [43](#), [45](#)

Goldberg, D. H. **Genetic algorithms in search, optimisation, and machine learning**. Mass: Addison-Wesley Publishing, 1989. 412 p. [63](#), [67](#)

Lacerda, E. G. M.; Carvalho, A. C. P. L. F. Introdução aos algoritmos genéticos. In: Congresso Nacional da Sociedade Brasileira de Computação, 19. **Anais**. Rio de Janeiro (RJ): Edições Entrelugar, 1999. v. 2, p. 51–126. [65](#)

Lee, C. C. Fuzzy logic in control systems: fuzzy logic controller - part I. **IEEE Transactions on Systems, Man and Cybernetics**, v. 25, n. 4, p. 404–418, 1990. [19](#), [35](#)

Mamdani, E. H.; Assilan, S. Applications of fuzzy algorithm for control of simple dynamic plant. **Proceedings of the IEEE**, v. 121, n. 12, p. 1585–1888, 1974. [20](#), [33](#)

Michalewicz, Z. **Genetic algorithms + data structures = evolution programs**. Berlin: Springer-Verlag, 1996. 387 p. [67](#)

Pedrycz, W. **Computational intelligence - an introduction**. New York: CRC Press, 1998. 284 p. [63](#)

Pedrycz, W.; Gomide, F. **An introduction to fuzzy sets: analysis and design**. London: MIT Press, 1998. 465 p. [23](#), [31](#), [34](#)

Roger, J. J. S.; Gulley, N. **Fuzzy logic toolbox: for use with MATLAB**. Mass: The MathWorks, 1995. [20](#)

Ruspini, E. A new approach to clustering. **Information and Control**, v. 15, n. 1, p. 22–32, 1969. [33](#)

Ruspini, E.; Bonissone, P.; Pedrycz, W. **Handbook of fuzzy computation**. London: IOP Publishing Ltd, 1998. 504 p. [28](#), [31](#)

Sandri, S.; Correa, C. Lógica nebulosa. In: Escola de Redes Neurais - ITA, 5. **Anais.** São José dos Campos (SP): CTA/ITA, 1999. v. 1, p. c073–c090. [19](#), [24](#), [34](#), [35](#)

Takagi, T.; Sugeno, T. Fuzzy identification of systems and its application to modeling and control. **IEEE Transactions on Systems, Man and Cybernetics**, v. 15, n. 1, p. 116–132, 1985. [31](#)

Tanaka, K. **An introduction to fuzzy logic for practical applications.** New York: Springer, 1991. 136 p. [23](#)

Tsoukalas, L. H.; Uhrig, R. E. **Fuzzy and neural approaches in engineering.** New York: A. Wiley-Interscience Publication, 1997. 210 p. [23](#)

Weisbrod, J.; Fantana, N. L. Detecting local inconsistency and incompleteness in fuzzy rule bases. In: European Congress on Intelligent Techniques and Soft Computing (EUFIT'96), 4, Aachen (Germany), 1996. **Proceedings.** Aachen: ELITE foundation, 1996. v. 1, p. 656–660. [41](#)

Yen, J. Fuzzy logic - a modern perspective. **IEEE Transactions on Knowledge and Data Engineering**, v. 11, n. 1, Jan/Feb 1989. [23](#), [31](#)

Yu, W.; Bien, Z. Design of fuzzy logic controller with inconsistent rule base. **Journal of Intelligent and Fuzzy Systems**, v. 2, n. 1, p. 147–159, 1994. [31](#)

Zadeh, L. A. Fuzzy sets. **Fuzzy Sets, Information and Control**, v. 8, n. 3, p. 338–353, June 1965. [19](#), [23](#)