

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

Projeto Automático de Sistemas Nebulosos: Uma Abordagem Co-Evolutiva

Myriam Regattieri De Biase da Silva Delgado
Prof. Dr. Fernando José Von Zuben (Orientador)
Prof. Dr. Fernando Gomide (Co-orientador)

Tese apresentada à Pós-graduação da Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como requisito parcial à obtenção do grau de **Doutor em Engenharia Elétrica** na área de Engenharia de Computação.

Banca examinadora:

Profa. Dra. Marley Maria Bernardes Rebuszi Vellasco– DEE, PUC-Rio, RJ
Prof. Dr. Maurício Fernandes Figueiredo– DIN, UEM, PR
Prof. Dr. Ricardo Ribeiro Gudwin– DCA, FEEC, UNICAMP
Prof. Dr. Wagner Caradori do Amaral– DCA, FEEC, UNICAMP
Prof. Dr. Márcio Luiz Andrade Netto– DCA, FEEC, UNICAMP

Campinas, 26 de Fevereiro de 2002

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

D378p Delgado, Myriam Regattieri De Biase da Silva
Projeto automático de sistemas nebulosos: uma
abordagem co-evolutiva / Myriam Regattieri De Biase da
Silva Delgado. --Campinas, SP: [s.n.], 2002.

Orientadores: Fernando José Von Zuben e Fernando
Gomide.

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

1. Algoritmos genéticos. 2. Cooperação. 3. Hierarquias.
I. Von Zuben, Fernando José. II. Gomide, Fernando. III.
Universidade Estadual de Campinas. Faculdade de
Engenharia Elétrica e de Computação. IV. Título.

Resumo

Esta tese propõe uma abordagem co-evolutiva para solucionar o problema de projeto automático de sistemas nebulosos. A co-evolução permite que relações de hierarquia e cooperação sejam estabelecidas entre indivíduos representando diferentes parâmetros dos sistemas nebulosos. A abordagem proposta recorre a diferentes espécies que codificam soluções parciais do problema de projeto automático de sistemas nebulosos e estão organizadas em quatro níveis hierárquicos. Cada nível hierárquico codifica as funções de pertinência, as regras individuais, as bases de regras e os sistemas nebulosos, respectivamente. Um esquema especial de avaliação de *fitness* é proposto para estimar o desempenho de cada indivíduo das diferentes espécies. Restrições e objetivos locais devem ser observados em todos os níveis, de modo a garantir a ocorrência de indivíduos caracterizados pela simplicidade das regras nebulosas, compactação e consistência da base de regras e visibilidade na partição do universo. A abordagem permite a evolução de modelos Mamdani ou Takagi-Sugeno. Além de melhorar o desempenho em termos de acuidade e interpretabilidade, a abordagem co-evolutiva aumenta a autonomia do projeto ao minimizar a intervenção do usuário, uma vez que permite o ajuste automático de uma série de parâmetros críticos, como tipo e total das regras nebulosas, variáveis relevantes (tanto para a regra quanto para a aplicação), tipo e localização das funções de pertinência, operador de agregação dos antecedentes e, para os modelos Mamdani, operador de agregação, semântica das regras e o método de obtenção de saídas não-nebulosas. O desempenho da abordagem proposta é avaliado através de problemas de aproximação de funções e classificação de padrões.

Abstract

This thesis proposes a co-evolutionary-based approach to solve the problem of automatic fuzzy system design. The co-evolution supports hierarchical and collaborative relations among individuals representing different parameters of fuzzy models. The proposed approach takes species which encode partial solutions to fuzzy modeling problems, organized into four hierarchical levels. Each hierarchical level encodes membership functions, individual rules, rule-bases and fuzzy systems, respectively. A special fitness evaluation scheme is proposed to measure the performance of each individual of different species. Constraints and local objectives must be observed at all hierarchical levels to guarantee the occurrence of individuals characterized by the simplicity of fuzzy rules, rule compactness, rule base consistency and visibility in the universe partition. The approach allows

the evolution of Mamdani or Takagi-Sugeno fuzzy models. In addition to performance improvement in terms of accuracy and interpretability, the co-evolutionary approach increases autonomy by minimizing user intervention, since it allows automatic tuning of a number of critical parameters, like type and total of fuzzy rules, relevant variables (for each rule and for the whole application), shape and location of membership functions, antecedent aggregation operator, and, for Mamdani models, aggregation operator, rule semantic, and the defuzzification method. The performance of the approach is evaluated via function approximation and pattern classification problems.

Para Armando e Carolina, com amor.

Agradeço

Ao meu marido Armando e à minha filha Carolina, por serem minha força e fonte de inspiração diária e por terem compreendido a minha ausência em tantas ocasiões em que precisei me dedicar integralmente ao trabalho.

Ao meu orientador e amigo prof. Fernando Von Zuben, por ter acreditado tanto neste trabalho, pela dedicação e competência demonstradas, que fazem dele um modelo a ser seguido.

Ao meu co-orientador prof. Fernando Gomide, pelas valiosas discussões, sugestões e pela ampla divulgação da proposta e dos resultados obtidos.

À minha mãe Anastasia e irmãos Waltinho e Wania, por me proporcionarem a base emocional necessária para vencer as etapas mais complicadas do meu processo de aprendizado.

Ao meu pai Walter (*in memoriam*), que será sempre fonte de orgulho, admiração e uma referência para o meu crescimento pessoal e profissional.

À D. Jupyra por todo o seu amor e pelas fortes orações que me ajudaram a vencer uma série de barreiras.

Aos amigos de longa data Ricardo e Luciana, Gonzaga e Jussara, pelo carinho e força que sempre me passaram.

A todos os amigos do laboratório, que fazem da FEEC/Unicamp um ambiente sempre agradável e descontraído para se trabalhar.

Aos membros da banca examinadora pelas interessantes sugestões apresentadas.

Ao CEFET-PR e à Capes-PICDT pelo suporte financeiro que possibilitou a realização deste trabalho.

Conteúdo

1	Introdução	1
2	Computação Evolutiva	9
2.1	Inspiração Biológica	9
2.1.1	Terminologia Biológica	10
2.2	As Técnicas da Computação Evolutiva	11
2.2.1	A Computação Evolutiva como Método de Busca	12
2.2.2	Algoritmos Evolutivos	13
2.2.3	Espaço Fenotípico e Espaço Genotípico	15
2.3	Algoritmos Genéticos	19
2.3.1	Codificação	22
2.3.2	Métodos de Seleção	23
2.3.3	Operadores Genéticos	27
2.3.4	Exemplo de Aplicação	32
2.4	Co-Evolução	35
3	Sistemas Nebulosos	41
3.1	Introdução aos Conjuntos Nebulosos	43
3.1.1	Funções de Pertinência	44
3.1.2	Definições Básicas em Conjuntos Nebulosos	47
3.1.3	Operações com Conjuntos Nebulosos	48
3.1.4	Relações Nebulosas	54
3.2	Computação com Regras Nebulosas: Raciocínio Aproximado	57
3.2.1	Variáveis Lingüísticas	58
3.2.2	Regras Nebulosas	60

3.2.3	Raciocínio Aproximado	63
3.2.4	Regra Composicional de Inferência	64
3.2.5	O Processo de Inferência no Raciocínio Aproximado	66
3.3	Sistemas Nebulosos	68
3.3.1	Métodos de Transformação da Saída	70
3.3.2	Mecanismos de Inferência	71
3.3.3	Modelos de Sistemas Nebulosos	75
4	Abordagem Co-Evolutiva para o Projeto Automático de Sistemas Nebulosos	81
4.1	Projeto Automático de Sistemas Nebulosos	83
4.2	Abordagens Relacionadas	85
4.3	Sistema Nebuloso Co-Evolutivo: Uma Proposta Baseada em Hierarquia e Cooperação	90
4.3.1	A Abordagem Co-evolutiva	90
4.3.2	Hierarquia e Cooperação	93
4.3.3	Algoritmo Genético Proposto	97
4.3.4	Operadores Evolutivos	100
4.3.5	Avaliação do <i>Fitness</i>	103
4.4	O Modelo do Sistema Nebuloso com Conseqüentes Não-Lineares	105
4.4.1	Exemplo de Utilização do Modelo TS com Conseqüentes Não-Lineares	106
4.4.2	Otimização do Conseqüente: Método Global	107
4.4.3	Otimização do Conseqüente: Método Local	110
4.4.4	O Processo de Poda	110
5	Simulações e Resultados	115
5.1	Abordagens de Comparação	116
5.1.1	Considerações Gerais	118
5.2	Aproximação de Funções: Dados sem Ruído	121
5.2.1	Aproximação do Mapeamento F_1 : Análise de Acuidade	122
5.2.2	Aproximação do Mapeamento F_2 : Análise de Interpretabilidade	125
5.3	Aproximação de Funções: Dados com Ruído	129
5.3.1	Erro de Previsão	130
5.3.2	Aproximação no \mathfrak{R}^2 Usando Dados com Ruído	131
5.3.3	Aproximação no \mathfrak{R}^3 Usando Dados com Ruído	136

5.4	Problemas de Classificação	139
5.4.1	Dados das Espirais Concêntricas	139
5.4.2	Dados da IRIS	143
5.5	Experimentos de Caminho Fechado	146
6	Conclusões	151
6.1	Perspectivas Futuras	154
	Glossário	157
	Bibliografia	167
	Índice de Autores	177
	Índice de Assuntos	179

Lista de Tabelas

3.1	Propriedades das Operações com Conjuntos Clássicos	50
4.1	Comparação das Abordagens para o Projeto Automático de Sistemas Nebulosos	91
4.2	Parâmetros do Modelo Takagi-Sugeno Exemplificado	107
5.1	Resultados das Simulações: Problema de Aproximação do Mapeamento F_1	123
5.2	Custo Computacional no Problema de Aproximação do Mapeamento F_1	125
5.3	Resultados das Simulações: Problema de Aproximação do Mapeamento F_2	127
5.4	Parâmetros do Modelo Takagi-Sugeno Obtido	127
5.5	Componentes de Limiar e Variância do Erro	132
5.6	Resultados das Simulações: Problema de Aproximação do Mapeamento F_3	135
5.7	Resultados das Simulações: Problema de Aproximação do Mapeamento F_4	137
5.8	Resultados das Simulações: Problema das Espirais	142
5.9	Resultados das Simulações: Problema da IRIS	144
5.10	Custo Computacional no Problema da IRIS	146
5.11	Parâmetros dos Modelos Takagi-Sugeno nos Problemas de Caminho Fechado	148

Lista de Figuras

2.1	Fluxograma dos Algoritmos Evolutivos	14
2.2	Evolução da População (Genótipo \times Fenótipo)	16
2.3	Avaliação do Desempenho Médio da População	18
2.4	Esquema Geral de Evolução de um Algoritmo Genético	21
2.5	Exemplo de Roleta na Seleção Proporcional ao Desempenho	24
2.6	Exemplo de Operadores de Crossover	28
2.7	Exemplo de Operadores de Mutação	31
2.8	Problema de Otimização da Caixa Preta	32
2.9	Codificação Binária para o Problema da Caixa Preta	33
2.10	Evolução do Desempenho no Problema da Caixa Preta	34
2.11	Exemplo de Hierarquia na Evolução de duas Populações: Neurônios e Protótipos de Redes	37
2.12	Exemplo de Co-evolução Cooperativa Proposta por Potter & De Jong (2000)	40
3.1	Funções de Pertinência em Conjuntos Clássicos e Nebulosos	45
3.2	Exemplos de Formatos das Funções de Pertinência	46
3.3	Diferentes Informações em Conjuntos Nebulosos	48
3.4	Convexidade e Normalidade em Conjuntos Nebulosos	49
3.5	Operações com Conjuntos Nebulosos	51
3.6	Exemplo de Agregação por Normas e Co-normas Triangulares	53
3.7	Exemplo de Relação Nebulosa num Universo Contínuo	55
3.8	Exemplo de Relação Nebulosa num Universo Discreto	55
3.9	Composição de Relações Nebulosas	57
3.10	Exemplo de Partições do Universo de Discurso	59
3.11	Interpretação da Regra Nebulosa	62
3.12	Processo de Inferência com Funções	64

3.13	Processos de Inferência	66
3.14	Exemplo de Raciocínio Aproximado: Representação Plano	67
3.15	Exemplo de Raciocínio Aproximado: Representação no Espaço	69
3.16	Raciocínio Aproximado Envolvendo uma Base de Regras	70
3.17	Inferência no Modelo Mamdani	77
3.18	Inferência no Modelo Takagi-Sugeno	78
4.1	Análise de Interpretabilidade das Partições Nebulosas	84
4.2	O Modelo Co-evolutivo Proposto	92
4.3	Esquema de Codificação dos Sistemas Nebulosos no Nível IV	93
4.4	Esquema de Hierarquia e Cooperação	95
4.5	Codificação das Funções de Pertinência Trapezoidal e Triangular	96
4.6	Codificação das Funções Gaussianas	97
4.7	Algoritmo Genético do Modelo Co-Evolutivo Proposto	98
4.8	Esquema de Avaliação do Fitness	104
4.9	Partição Nebulosa do Modelo Takagi-Sugeno Exemplificado	107
4.10	Mapeamento Produzido pelo Modelo TS Exemplificado	108
4.11	Algoritmo Proposto para o Processo Poda	112
5.1	Modelo Evolutivo de Comparação	117
5.2	Mapeamento F_1 e Erros de Aproximação	124
5.3	Evolução dos Erros de Treinamento e Teste	125
5.4	Mapeamento F_2 e Resultados das Aproximações	128
5.5	Partições Nebulosas na Aproximação de F_2	129
5.6	Análise Gráfica dos Componentes de Limiar e Variância do Erro	133
5.7	Influência do Grau de Sobreposição no Erro	134
5.8	Mapeamento F_3 e Resultados de Aproximação	136
5.9	Partições Nebulosas na Aproximação do Mapeamento F_3	137
5.10	Evolução do Erro na Aproximação do Mapeamento F_4	138
5.11	Aproximação do Mapeamento F_4	138
5.12	Partições Nebulosas na Aproximação de F_4	139
5.13	Dados de Treinamento no Problema das Espirais	140
5.14	Evolução do Erro de Classificação no Problema das Espirais	141
5.15	Desempenho dos Classificadores Nebulosos no Problema das Espirais	142
5.16	Partição Nebulosa no Problema das Espirais	143

5.17	Partição Nebulosa no Problema de Classificação da IRIS	145
5.18	Partições Nebulosas para os Problemas de Caminho Fechado	149
5.19	Resultados das Aproximações: Problemas de Caminho Fechado	150

Capítulo 1

Introdução

A solução de problemas práticos de grande complexidade requer a construção de sistemas híbridos, capazes de integrar coerentemente técnicas e metodologias originárias de diferentes áreas de pesquisa. A Computação Flexível (*Soft Computing*) engloba três diferentes ramos: Computação Evolutiva, Redes Neurais e Sistemas Nebulosos, e se apresenta como uma alternativa promissora para a obtenção de sistemas computacionais híbridos (Bäck *et al.*, 2000a,b; Lin & Lee, 1996; Jang *et al.*, 1997; Zadeh, 1965). Os diferentes paradigmas computacionais que formam a Computação Flexível (CF) permitem a simulação, ainda que sujeita a inúmeras simplificações, da capacidade de raciocínio e aprendizado do ser humano em ambientes caracterizados por incerteza e imprecisão (Jang *et al.*, 1997).

A Computação Evolutiva (CE), que pode ser definida como a simulação de mecanismos de evolução através do uso de computadores, possibilitou o surgimento de uma nova filosofia de máquinas inteligentes. O conceito de inteligência aqui adotado tem por base a proposta de Fogel (1995), que define inteligência como a capacidade de um sistema adaptar seu comportamento para atingir seus objetivos numa variedade de situações. Sob este ponto de vista, inteligência e evolução são dois conceitos intimamente relacionados. A simulação de processos evolutivos biológicos permite o entendimento de como a evolução guia os seres vivos na direção de um maior nível de inteligência (Fogel, 1995). A evolução, caracterizada por diferentes níveis de representação (gene, cromossomo, indivíduo, espécie, ecossistema), converge ao final do processo para soluções que apresentam um comportamento mais adaptado ao ambiente. A modelagem do processo evolutivo emprega uma série de algoritmos de otimização baseados em regras simples e implementados como métodos de busca em superfícies de resposta de desempenho (*fitness surface*). O processo de otimização, inerente à seleção dos elementos mais bem adaptados, melhora a qualidade média

das soluções obtidas ao longo das gerações. Entretanto, o grande potencial dos algoritmos baseados em computação evolutiva vem da integração da ampla exploração do espaço de busca com um processo de busca mais localizada, denominado exploração. Esta integração resulta num alto grau de robustez, que permite a aplicação de CE em vários problemas práticos, junto aos quais outras estratégias de solução se mostram inócuas.

Já os paradigmas baseados em Redes Neurais (RNs) modelam o cérebro como um sistema de processamento de informação em que podem ser destacadas suas propriedades dinâmicas, distributividade, não-linearidade e auto-organização (Haykin, 1999). Algumas características básicas justificam a inspiração em mecanismos conexionistas: as redes neurais biológicas executam tarefas complexas, através da interação de elementos básicos (neurônios), que executam operações relativamente simples; e mais, estes sistemas reúnem a característica de tolerância a falhas presentes tanto no ambiente como em seus elementos básicos (Berthold & Hand, 1999). As redes neurais artificiais são estruturas de processamento paralelo de informação, compostas por unidades básicas altamente interconectadas (neurônios artificiais ou nodos). Os modelos de redes neurais artificiais apresentam propriedade de representação distribuída, i.e., o comportamento coletivo da rede é o responsável pelo seu poder computacional e nenhum neurônio individual armazena informação específica. Cada neurônio responde ao estímulo de entrada e sua ativação depende completamente de informações locais. O conhecimento é armazenado nas conexões da rede, sendo que a cada uma é associado um peso ou valor numérico, embora processamento simbólico também possa estar presente (Rocha, 1992). Através do ajuste destes pesos é possível a implementação de processos de aprendizado. As redes neurais artificiais recuperam, de forma rápida, a informação armazenada e apresentam capacidade de generalização, ou seja, respondem de maneira correta para certos estímulos não apresentados na fase de treinamento (Lin & Lee, 1996).

Por sua vez, os Sistemas Nebulosos (SNs), baseados na Teoria dos Conjuntos Nebulosos introduzida por Zadeh (1965), têm um papel fundamental nas mais diversas áreas do processamento de informação. Os conjuntos nebulosos representam uma ponte de ligação entre processamento simbólico e numérico. A teoria de sistemas nebulosos utiliza símbolos (termos lingüísticos) com os quais estão associadas semânticas bem definidas que, após serem convertidas em funções de pertinência de conjuntos nebulosos, possibilitam o processamento numérico destes símbolos ou conceitos (Pedrycz & Gomide, 1998). Em sistemas baseados em regras nebulosas, informações imprecisas e incompletas, em analogia às informações sensoriais recebidas pelos órgãos e interpretadas pelo cérebro, são tratadas de forma lingüística (Jang *et al.*, 1997). Segundo Jang *et al.* (1997), a estrutura básica de um sistema nebuloso possui três componentes conceituais: uma **base**

de regras, que contém o conjunto de regras nebulosas; uma **base de dados**, que define as funções de pertinência usadas nas regras nebulosas e um **mecanismo de raciocínio**, que realiza um procedimento de inferência (raciocínio nebuloso) para derivar a saída ou conclusão, baseado nas regras e fatos conhecidos. A seleção de uma base de regras nebulosas, associada à escolha de semânticas e operadores apropriados, abre caminho para a criação de sistemas nebulosos que modelam e processam o conhecimento do especialista em aplicações específicas.

De uma forma geral, a obtenção da solução de um problema depende de um processo de busca dentro de um espaço de soluções potenciais. Por exemplo, quando a tarefa é a determinação da estrutura de um sistema nebuloso, o espaço de busca que representa os possíveis modelos é extremamente complexo. Isto porque envolve a definição de diferentes parâmetros relacionados à base de regras, base de dados e aos mecanismos de inferência (raciocínio nebuloso). A definição manual destes parâmetros exige o conhecimento do especialista, o que restringe o universo de aplicações. A solução do problema da definição automática dos parâmetros dos SNs, conhecida como projeto automático de sistemas nebulosos, requer a implementação de sistemas computacionais mais elaborados.

Este trabalho apresentará uma abordagem para o projeto de sistemas nebulosos na qual um grande número de parâmetros do modelo nebuloso será obtido de forma automática. Na proposta a ser apresentada, a base de regras (total e tipo das regras nebulosas), a maioria dos parâmetros relacionados à base de dados (formato e localização ou suporte das funções de pertinência) e os operadores associados ao mecanismo de raciocínio ou inferência (operadores de agregação do antecedente, etc.), deverão ser obtidos pela abordagem proposta, sem a necessidade de intervenção do usuário ou projetista.

A despeito do potencial individual de cada uma das técnicas que formam o alicerce da computação flexível (CF), a grande perspectiva do uso de CF em problemas práticos diz respeito à integração destas técnicas. O uso conjunto dos diferentes paradigmas computacionais da CF permite a exploração da capacidade de aprendizado e adaptação associada às redes neurais, a representação do conhecimento através de uma base de regras nebulosas, característica dos sistemas nebulosos e, por fim, a obtenção de soluções por meio de eficientes métodos de busca exploratória e exploratória, presentes em algoritmos baseados em computação evolutiva (Jang *et al.*, 1997). A literatura apresenta inúmeras propostas que empregam duas ou mais técnicas da computação flexível de forma colaborativa (Cordón & Herrera, 1999; Delgado *et al.*, 2001a; Hoffmann & Nelles, 2000; Jang, 1993; Karr, 1991b; Lin & Lee, 1996; Moriarty & Miikkulainen, 1998; Nauck & Kruse, 1999; Yao, 1993). Neste contexto, destaca-se a abordagem neuro-genética proposta por Moriarty & Miikku-

lainen (1998), que utiliza a CE para a obtenção de modelos de RNs. Outro trabalho que merece destaque é o sistema neuro-nebuloso proposto por Jang (1993), que é conhecido como ANFIS e emprega RNs na obtenção dos parâmetros de SNs. Na abordagem a ser proposta neste trabalho, a CE é utilizada para a otimização dos parâmetros de SNs.

A justificativa para a integração das diferentes técnicas da CF tem por base as desvantagens da utilização isolada destas técnicas na solução de problemas complexos, já que existe uma grande diversidade de parâmetros e aspectos estruturais vinculados aos modelos baseados em computação flexível. Como um exemplo, considere o caso das redes neurais artificiais. Embora a arquitetura *perceptron* multicamadas apresente capacidade de aproximação universal (Hornik *et al.*, 1989), não há procedimentos sistemáticos de especificação direta do número de neurônios que devem compor as camadas intermediárias da rede neural. Além disso, é sabido que o número adequado de neurônios é diferente para cada aplicação. Sendo assim, procedimentos automáticos de definição da dimensão e topologia das redes neurais do tipo *perceptron* multicamadas são desejados e deverão estar associados a processos iterativos de busca. Nestes casos, o uso de técnicas evolutivas, mostra-se promissor (Yao, 1993). No entanto, vale salientar que, devido à necessidade de treinamento das redes a cada geração, a tratabilidade computacional do sistema pode ficar comprometida. Embora as redes neurais tenham sido utilizadas acima como exemplo, o mesmo se aplica aos sistemas nebulosos, junto aos quais está posicionado o foco de atenção desta tese.

Além dos problemas de modelagem e convergência associados aos algoritmos de treinamento, o estigma da “caixa-preta” pode comprometer o uso das abordagens baseadas em redes neurais, em algumas aplicações. Isto porque a representação por um conjunto de conexões torna difícil o entendimento e análise do conhecimento mapeado ao final do processo de aprendizado. Em contrapartida, sistemas nebulosos usam regras de inferência para descrever o conhecimento do especialista. No entanto, em muitos casos, este conhecimento pode não estar disponível e o ajuste dos parâmetros do sistema nebuloso pode se mostrar bastante complexo.

Como o processo de integração de metodologias de CF deve considerar as peculiaridades de cada aplicação, não existe uma abordagem sistemática para o projeto automático de sistemas híbridos de CF. Várias propostas têm sido apresentadas neste sentido, sendo que esta tese apresenta contribuições especificamente voltadas para o projeto automático de sistemas nebulosos usando uma abordagem co-evolutiva.

A seqüência deste capítulo se ocupa em apresentar o estado da arte de sistemas nebulosos híbridos. A pesquisa nesta área concentra esforços de pesquisadores atuando em diferentes frentes (Cordón & Herrera, 1999; Delgado *et al.*, 2000a; Hoffmann & Nelles, 2000; Jang, 1993; Kosko,

1997; Lee & Takagi, 1993; Lin & Lee, 1996; Shi *et al.*, 1998; Takagi & Lee, 1993).

Em um projeto genérico de sistemas nebulosos, é importante se definir um critério para medir a eficiência das abordagens propostas. Neste trabalho, a eficiência do projeto será avaliada com base em três aspectos distintos:

- *acuidade da solução obtida*: dado um problema de aproximação qualquer e considerando-se diferentes técnicas de projeto de sistemas nebulosos, não é possível antecipar qual das abordagens irá obter o melhor desempenho em termos do erro, não importando a eficiência do algoritmo de aprendizado e a adequação das condições iniciais (Wolpert & Macready, 1997). Portanto, o critério de erro deve ser considerado, mesmo que associado a outros critérios, na escolha da melhor solução em cada caso.
- *interpretabilidade da solução obtida*: a justificativa para o uso de sistemas nebulosos, em contrapartida a outras técnicas, como as redes neurais artificiais, recai sempre no fato destes sistemas produzirem soluções interpretáveis, em maior ou menor grau, por um operador humano. Portanto, além do critério de erro, a facilidade com que a solução encontrada é compreendida e interpretada também deve ser considerada. Nesta proposta, a interpretabilidade será medida com base em 4 critérios distintos: visibilidade da partição nebulosa, simplicidade das regras, compactação da base de regras e consistência de cada regra que a compõe.
- *automatização do processo de obtenção da solução*: uma abordagem automática pressupõe um grau de intervenção mínimo por parte do usuário na definição dos parâmetros críticos do modelo a ser produzido. Portanto, este é mais um critério a ser considerado na escolha da abordagem mais apropriada.

Apesar das limitações já antecipadas, as RNs aparecem como uma alternativa para o projeto de sistemas nebulosos, pois permitem que parte do conhecimento do especialista seja obtida via treinamento, resultando em sistemas neuro-nebulosos (*neuro-fuzzy*) com alto poder de adaptação (Jang, 1993; Hang, 1997). Um exemplo bastante difundido é o sistema híbrido conhecido como ANFIS (Jang, 1993), que consiste na estruturação da base de regras nebulosas por meio de uma rede neural artificial. Embora seja considerado uma referência em qualquer aplicação envolvendo aproximação de funções, o ANFIS apresenta algumas desvantagens: como discutido por Rizzi *et al.* (1999), o uso do ANFIS é restringido pela quantidade de parâmetros a serem definidos pelo usuário, comprometendo a automatização do projeto e, conforme apresentado por Chen &

Liou (1999), o problema dos mínimos locais, resultante do uso do método de otimização baseado no gradiente pode comprometer a eficácia do processo de aprendizado do ANFIS em algumas aplicações específicas.

O uso de computação evolutiva para o ajuste de parâmetros de sistemas nebulosos tem sido visto como uma solução bem interessante. Abordagens baseadas em computação evolutiva, especialmente os algoritmos genéticos (AGs), produzem ferramentas poderosas para a otimização dos parâmetros de sistemas nebulosos, particularmente quando a parte mais significativa das decisões pode ser codificada em representações cromossômicas. O projeto automático de sistemas nebulosos utilizando algoritmos genéticos traz inúmeras vantagens:

- geralmente, a busca no espaço de soluções (estrutura e parâmetros do sistema nebuloso) é realizada de forma mais eficiente quando comparada com métodos tradicionais: buscas baseadas em AGs incluem elementos aleatórios e uma forma de paralelismo que permitem que se escape de mínimos locais ruins;
- as operações realizadas não requerem informações de derivadas parciais relacionadas, responsáveis pela indicação direta do quanto varia o desempenho do sistema para uma variação em cada parâmetro (Lee & Takagi, 1993; Michalewicz, 1996; Shi *et al.*, 1999). Com isso, é possível evoluir aspectos estruturais dos sistemas nebulosos.

A área que agrega os sistemas nebulosos e algoritmos genéticos é conhecida como *Genetic Fuzzy Systems* (GFS) e tem atraído a atenção de diversos pesquisadores (Cordón *et al.*, 2001c; Delgado *et al.*, 2001a; Heider & Drabe, 1997; Hoffmann & Pfister, 1995; Ishibuchi & Nakashima, 1999; Lin & Lee, 1996; Shi *et al.*, 1998; Siarry & Guely, 1998).

A maioria das propostas de GFS apresentadas na literatura utiliza apenas um nível de representação da solução, ou seja, representam a solução completa por apenas um cromossomo, tratando em um mesmo nível aspectos distintos do problema. Em tomadas de decisões complexas, como definição da estrutura e parâmetros de um sistema nebuloso, apenas um nível de representação pode ser inadequado. Nestes casos, a otimização de um grande número de parâmetros codificados em níveis distintos é certamente mais indicada (Delgado *et al.*, 1999; Heider & Drabe, 1997). Entretanto, abordagens mais complexas podem requerer esquemas de evolução elaborados, que incluam aspectos de co-evolução. Segundo Potter & De Jong (2000), para se utilizarem algoritmos genéticos de forma eficiente em problemas complexos, noções explícitas de modularidade devem ser introduzidas. Esta modularidade cria a oportunidade das soluções evoluírem na forma de sub-componentes, interagindo de maneira co-adaptativa. O modelo co-evolutivo proposto por Potter

& De Jong (2000), será tomado como ponto de partida para o desenvolvimento das ferramentas co-evolutivas a serem propostas nesta tese.

A proposta da solução do problema de projeto automático de sistemas nebulosos a ser apresentada neste trabalho é baseada em co-evolução, com componentes de hierarquia e cooperação. Populações de diferentes níveis hierárquicos significam espécies distintas, que codificam soluções parciais do problema de projeto de sistemas nebulosos. A estrutura hierárquica pode ser implementada de modo que indivíduos dos quatro níveis distintos - partições nebulosas (nível I), regras individuais (nível II), base de regras (nível III) e sistemas nebulosos (nível IV) - mantenham uma relação cooperativa e o *fitness* de um indivíduo dependa do *fitness* de indivíduos pertencentes aos outros níveis.

A abordagem permite a obtenção de modelos de sistemas nebulosos do tipo Mamdani ou Takagi-Sugeno (TS). A intervenção do usuário é mínima pois o processo co-evolutivo atua sobre parâmetros críticos como:

- tipo e total das regras nebulosas;
- variáveis relevantes tanto para as regras nebulosas quanto para a aplicação;
- tipo e localização das funções de pertinência;
- operador de agregação dos antecedentes;
- e para os modelos Mamdani,
 - semântica das regras;
 - operador de agregação das regras;
 - método de obtenção de saídas não-nebulosas (*defuzzification*).

Os modelos TS (Takagi & Sugeno, 1983) serão enfatizados pois a sua adoção, além de permitir a modelagem de sistemas complexos com um menor número de regras, possibilita o uso de algoritmos de estimação eficientes para a obtenção dos parâmetros dos consequentes (Delgado *et al.*, 2000b; Takagi & Lee, 1993). Neste sentido, dois métodos de otimização dos parâmetros do consequente TS serão avaliados: o método de otimização global baseado nos quadrados mínimos e o método de otimização local dos quadrados mínimos ponderados. Serão também testadas diferentes classes de funções no consequente, variando da classe mais simples (constante) para as mais complexas (não-linear). Um método de poda será proposto para se eliminar o excesso de flexibilidade resultante de possíveis redundâncias nas regras nebulosas.

Um outro aspecto que será discutido neste trabalho diz respeito ao alto custo computacional associado ao projeto automático de sistemas nebulosos (Delgado *et al.*, 2001a). Algoritmos de ajuste (aprendizado), nos quais tanto a estrutura do sistema nebuloso como seus parâmetros são modificados, dão origem a uma classe poderosa de dispositivos não-paramétricos. Dispositivos de aprendizado não-paramétricos representam um novo paradigma na modelagem de sistemas (Härdle, 1990). Neste novo paradigma, existe uma tendência de se incrementar o custo da modelagem com o objetivo de se obter um solução mais parcimoniosa ao final do processo. Isto é exatamente o que se espera em aplicações práticas.

Este trabalho está estruturado em 6 capítulos. Após esta introdução, o capítulo 2 apresentará as bases da teoria de Computação Evolutiva. No capítulo 3, os fundamentos teóricos dos sistemas nebulosos serão discutidos. A proposta de projeto automático de sistemas nebulosos baseada em co-evolução será descrita no capítulo 4. Os resultados de simulações, nos mais diversos domínios de aplicação, serão detalhados no capítulo 5. Finalmente, o capítulo 6 descreverá as conclusões do trabalho e perspectivas futuras.

Capítulo 2

Computação Evolutiva

Os sistemas baseados em computação evolutiva têm sido muito utilizados como ferramentas de otimização em problemas de engenharia. Tecnicamente, sistemas adaptativos para a solução de problemas complexos necessitam de algum tipo de operador de busca iterativa. Este processo de busca deve explorar o espaço que descreve todas as configurações possíveis de solução e os sistemas evolutivos são uma alternativa interessante para realizar este procedimento.

Este capítulo apresenta os princípios básicos da computação evolutiva, com ênfase nos algoritmos genéticos. Na seção 2.1, as características que justificam a inspiração em mecanismos evolutivos biológicos são discutidas. A seção 2.2 traz as técnicas que compõem a computação evolutiva, analisa a importância destas técnicas como métodos efetivos de busca em espaços complexos, apresenta noções gerais envolvendo os algoritmos que implementam as técnicas da computação evolutiva, conhecidos como algoritmos evolutivos e, por fim, analisa a importância das representações genotípicas e fenotípicas dos indivíduos em uma população. A seção 2.3 discute os conceitos específicos dos algoritmos genéticos, descrevendo suas formas de representação, métodos de seleção e os operadores genéticos utilizados, além de trazer um exemplo de aplicação. Finalmente, a seção 2.4 aborda mecanismos evolutivos mais complexos, baseados nos princípios da co-evolução.

2.1 Inspiração Biológica

De forma genérica, as regras dos mecanismos evolutivos biológicos são relativamente simples: indivíduos ou espécies são submetidos a processos de variação aleatória (via mutação, crossover ou outros operadores), seguidos pela seleção natural, na qual o indivíduo melhor adaptado tende a sobreviver e se reproduzir, propagando seu material genético para futuras gerações. Embora sim-

ples, este mecanismo aparentemente é responsável pela extraordinária variedade e complexidade dos organismos encontrados na natureza (Mitchell, 1996).

Além do aspecto da simplicidade, a inspiração em mecanismos de evolução natural é justificada por uma série de características, dentre as quais destacam-se:

- **a busca da solução** - A evolução natural é um método de busca dentro de um enorme conjunto de candidatos à solução. Na biologia, isto é representado pelo conjunto de possíveis seqüências genéticas.
- **a adaptação**- Na natureza, há uma busca continuada por organismos que melhor se adaptem ao ambiente, o qual também sofre alterações. Como o ambiente também está sujeito a algumas modificações, a evolução pode ser vista como uma busca dentro de um conjunto de possibilidades que apresentam níveis de desempenho em constante alteração.
- **o paralelismo**- A evolução biológica é considerada um método de busca populacional, pois atua em múltiplos indivíduos (ou espécies) ao mesmo tempo.
- **a inovação**: A obtenção de soluções inovadoras devido, principalmente, à variação genética, também é uma característica da evolução natural.

Os sistemas baseados em CE buscam incorporar todas estas características.

2.1.1 Terminologia Biológica

A terminologia empregada na Computação Evolutiva tem forte inspiração biológica. Entretanto, é importante salientar que as estruturas computacionais são, invariavelmente, mais simples do que seus equivalentes biológicos. A seguir, serão detalhadas as definições biológicas que auxiliam no entendimento de algumas estruturas computacionais básicas utilizadas em Computação Evolutiva:

- **Cromossomo**: um cromossomo é uma estrutura nucleoprotéica formada por cadeias de DNA, sendo a base física dos genes do núcleo da célula. Toda célula de um mesmo organismo contém o mesmo conjunto de um ou mais cromossomos. Cada espécie apresenta um número característico de cromossomos.
- **Genes**: os genes são blocos funcionais de DNA, os quais codificam uma proteína específica. Cada gene está localizado em uma posição (*locus*) particular do cromossomo.
- **Alelos**: alelos são genes que se encontram no mesmo *locus* de dois cromossomos homólogos.

- **Organismos Diplóides:** quando os cromossomos são funcionalmente arranjados em pares (cada cromossomo proveniente de um dos pais), os respectivos organismos são chamados de diplóides.
- **Genoma:** conjunto de todos os cromossomos que compõem o material genético do organismo.
- **Crossover:** também conhecido como recombinação, consiste na troca (evento aleatório) de material genético entre dois cromossomos.
- **Mutação:** erro de cópia ou mudança aleatória no processo de transmissão do código genético dos indivíduos para os seus descendentes diretos.
- **Espécies:** populações de indivíduos geneticamente isoladas.

Tomando a definição de cromossomo, sua implementação em computador vai recorrer a estruturas de dados na forma de lista de atributos. Uma lista de atributos corresponde ao conjunto de especificações que caracteriza uma solução candidata. A partir desta abstração, todas as demais podem ser devidamente adaptadas para permitirem sua implementação computacional.

2.2 As Técnicas da Computação Evolutiva

A idéia de se utilizarem sistemas evolutivos na solução de problemas complexos remonta do início dos anos 60. As primeiras tentativas de se aplicar, de maneira formal, a teoria evolutiva em problemas práticos de engenharia foram realizadas nas áreas de controle de processos estocásticos, aprendizado de máquina e otimização de funções (Fogel, 1995).

Nos anos 60 e 70, Rechenberg (1965, 1973) apresentou as estratégias evolutivas (EEs), posteriormente desenvolvidas por Schwefel (1981). As EEs podem ser definidas como métodos para otimização de parâmetros com valores reais, propostas com o objetivo de imitar os princípios da evolução natural (variação genética), através de operadores de mutação específicos para produzir pequenas mudanças nestes parâmetros reais.

Em 1966, Fogel *et al.* (1966), desenvolveram a programação evolutiva (PE) que consiste em uma técnica na qual os candidatos à solução são representados por máquinas de estado finito. As máquinas de estado finito são transdutores que podem ser estimuladas por um alfabeto finito de símbolos de entrada, levando a um alfabeto finito de símbolos de saída, os quais também dependem

de um número finito de estados internos possíveis (Fogel, 1995). Estas máquinas são evoluídas por mutação aleatória de seus diagramas de transição de estado e seleção dos candidatos melhor adaptados.

Nos anos 70, Holland (1975) desenvolveu os algoritmos genéticos. Ao contrário das EEs e PE, o objetivo inicial dos algoritmos genéticos (AGs) não era a solução de problemas específicos e sim o estudo dos fenômenos de adaptação que ocorrem na natureza. Holland apresentou os AGs como uma abstração da evolução biológica. De seus estudos resultaram não só a possibilidade de simulação dos mecanismos da evolução natural em computadores, mas, principalmente, os fundamentos teóricos do processo de adaptação em AGs, sendo que o operador de recombinação ou crossover passou a ter um papel fundamental.

As estratégias evolutivas, juntamente com a programação evolutiva e os algoritmos genéticos, formam a base da computação evolutiva (Bäck *et al.*, 2000a,b). Nas implementações baseadas nas concepções originais, todas estas técnicas utilizam a seleção e empregam o operador de mutação, mas apenas os algoritmos genéticos fazem uso efetivo da recombinação. O operador de seleção escolhe, dentre os indivíduos da população, aqueles que irão se reproduzir. Em média, os indivíduos mais fortes (melhor adaptados) produzem mais descendentes do que aqueles mais fracos (menos adaptados). Na mutação, os valores dos elementos das estruturas codificadas são modificados aleatoriamente, a taxas reduzidas, permitindo que novas informações sejam incorporadas, à medida que o processo evolutivo avança e sem destruir os progressos já obtidos. Para as técnicas que utilizam crossover, as estruturas codificadas trocam partes de seus respectivos códigos, imitando o processo de crossover que ocorre na reprodução de organismos diplóides na natureza.

2.2.1 A Computação Evolutiva como Método de Busca

O processo de se procurar por uma solução desejada dentre uma coleção de candidatos é bastante comum em otimização de sistemas e recebe o nome de busca num espaço de soluções. Métodos tradicionais da teoria de otimização, conhecidos como *hill-climbing*, utilizam apenas um candidato à solução em cada instante e a busca é realizada através de modificações aplicadas a este candidato. Normalmente, uma nova solução é adotada apenas se seu desempenho é superior àquele da solução anterior. Esta restrição impede a ampla exploração do espaço de busca. Segundo Berthold & Hand (1999), de uma forma geral, a exploração não-determinística do espaço pode ser mais eficiente. Entretanto, métodos que fazem buscas totalmente aleatórias ignoram a existência de regiões promissoras, comprometendo assim a eficiência do processo. O uso da CE permite a construção de

poderosas ferramentas que exploram de forma paralela o espaço de busca com o aproveitamento das melhores soluções locais, representando, portanto, uma solução de compromisso.

Os mecanismos evolutivos se mostram particularmente apropriados para lidar com problemas computacionais nos quais o processo de busca da solução é realizado sobre um espaço multimodal, envolvendo um grande número de parâmetros. Estas buscas podem se beneficiar do uso efetivo do paralelismo, no qual diferentes possibilidades são exploradas simultaneamente e de forma eficiente (Mitchell, 1996). Esta característica também distingue os algoritmos baseados em CE de outras técnicas tradicionais, como *simulated annealing* (Aarts & Korst, 1989) e busca tabu (Glover & Laguna, 1997), que a cada iteração processam um único ponto no espaço.

O caráter não-determinístico dos métodos de busca baseados em evolução representa outro aspecto positivo. A vantagem advinda do processo de inovação causado por etapas aleatórias pode não ficar patente à primeira vista. Entretanto, mudanças aleatórias na codificação das soluções, em combinação com operadores de seleção baseados no desempenho dos indivíduos, dão origem a paradigmas de busca extremamente flexíveis. Buscas estocásticas aliadas a métodos de seleção e reprodução, que também podem ser não-determinísticos, permitem não só a exploração ampla do espaço como também buscas mais localizadas, denominadas de exploração. A ênfase em busca local na implantação de algoritmos genéticos possui ainda outras motivações, que são devidamente exploradas no contexto dos algoritmos meméticos (Moscato, 1999). Esta combinação de exploração e exploração confere um grande potencial para o uso da computação evolutiva na solução de problemas práticos.

2.2.2 Algoritmos Evolutivos

Paradigmas computacionais baseados em estratégias evolutivas, algoritmos genéticos e programação evolutiva, também conhecidos como algoritmos evolutivos (AEs), apresentam as seguintes características: mantêm um conjunto de soluções potenciais; apresentam um processo de seleção baseado no desempenho dos indivíduos; utilizam operadores genéticos no processo evolutivo.

Segundo Michalewicz (1996), um algoritmo evolutivo é um algoritmo probabilístico que mantém uma população de indivíduos $P^t = \{h_1^t, h_2^t, \dots, h_{S_p}^t\}$ a cada iteração t . Existem duas abordagens diferentes apresentadas na literatura: a abordagem *Pittsburgh* (mais usual), em que cada elemento da população ($h_i^t, i = 1, \dots, S_p$) corresponde a uma solução do problema; e a abordagem *Michigan*, em que toda a população P^t representa a solução do problema. Segundo a abordagem *Michigan*, cada indivíduo h_i^t representa uma solução parcial para o problema, que pode ser

implementada por diferentes estruturas de dados dependendo da técnica utilizada.

A figura 2.1 apresenta o fluxograma geral para desenvolvimento dos algoritmos evolutivos. Na abordagem *Pittsburgh*, cada solução h_i^t é avaliada por uma função de desempenho F , que mede o grau de adaptabilidade da solução potencial (*fitness*), dado por $F(h_i^t)$. Uma nova população (na iteração $t + 1$) é formada selecionando-se preferencialmente os indivíduos melhor adaptados. Alguns membros da nova população sofrem transformações através de operadores genéticos, resultando em novas soluções. Após atingir o critério de parada, o melhor indivíduo representa a solução a ser proposta (provavelmente próxima da solução desejada).

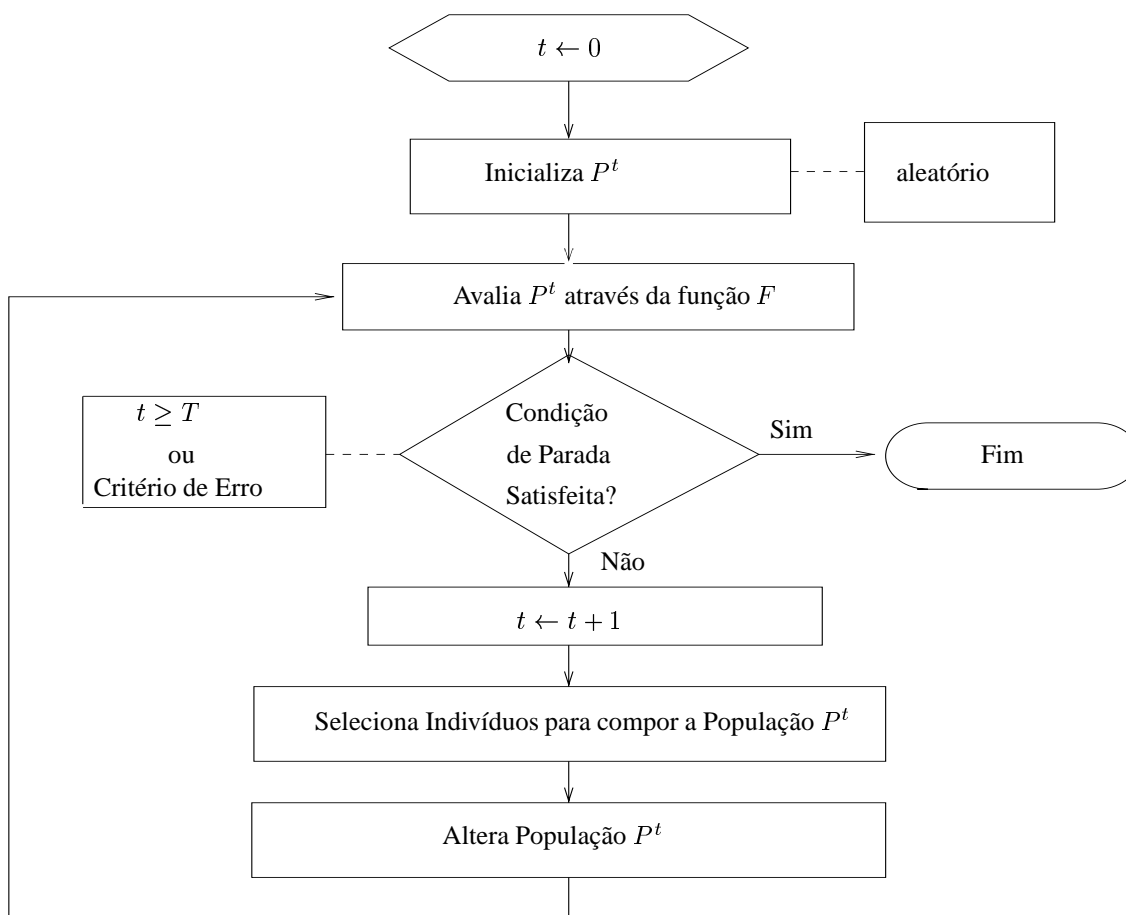


Figura 2.1: Fluxograma dos algoritmos evolutivos ($T =$ número máximo de gerações).

Embora apresentem inúmeras semelhanças, algumas características históricas distinguem entre si os algoritmos evolutivos: aqueles baseados nos AGs tradicionais evoluem uma população de soluções com codificação binária e aplicam tanto mutação quanto crossover para produzir novos indivíduos; os algoritmos inspirados na concepção original das EEs iniciam a evolução com apenas

dois indivíduos (o original e um derivado do processo de mutação) e a representação é feita através de vetores de números reais; os algoritmos baseados em PE utilizam apenas operadores de mutação e a codificação representa máquinas de estado finito.

Nos últimos anos, tem havido uma intensa integração entre pesquisadores das diferentes técnicas baseadas em computação evolutiva. Por exemplo, AGs com codificação real e operadores específicos para este tipo de representação; EEs com população de indivíduos e aplicação de operadores de crossover (Mitchell, 1996). Atualmente, é comum a utilização do termo algoritmo genético para descrever algoritmos evolutivos mais elaborados, independente de suas representações. Estas generalizações estão bem distantes da proposta original de Holland (1975), conhecida como algoritmo genético clássico.

2.2.3 Espaço Fenotípico e Espaço Genotípico

Cada indivíduo ou espécie pode ser visto como uma dualidade de seu genótipo e fenótipo (Fogel, 1995). O genótipo de um indivíduo representa o conjunto específico de genes do genoma (codificação genética). Já o fenótipo é a manifestação do genótipo no comportamento, fisiologia e morfologia do indivíduo, como um produto de sua interação com o ambiente. Em sistemas evolutivos naturais, não existe uma relação um-para-um entre o gene (elemento do genótipo) e a característica (elemento do fenótipo). Na evolução natural, duas características básicas impedem esta relação biunívoca entre genótipo e fenótipo: são elas, a pleiotropia e a poligenia, cujas definições são apresentadas a seguir.

- Pleiotropia: um único gene pode afetar diversos traços fenotípicos simultaneamente;
- Poligenia: uma característica fenotípica pode ser determinada pela interação de vários genes.

Sistemas obtidos por evolução natural são pleiotrópicos e poligênicos (Hartl & Clark, 1989). Entretanto, na maioria dos sistemas computacionais baseados em evolução estas duas características não estão presentes, uma vez que os efeitos da pleiotropia e poligenia tornam, geralmente, os resultados das variações genéticas imprevisíveis. Como forma de reduzir o custo computacional e possibilitar a análise das alterações genéticas, estes sistemas artificiais ainda adotam uma relação biunívoca entre o genótipo e o fenótipo de cada indivíduo da população (Von Zuben, 2000).

Conforme discutido por Atmar (1994), e Fogel (1995), o processo de evolução pode ser formalizado considerando-se dois espaços distintos: um espaço genotípico G , que representa a codificação genética; e um espaço fenotípico P , representando o comportamento do indivíduo frente ao ambiente em que está inserido. Considerando-se ainda um alfabeto de entrada I_{alf} composto de símbolos

$\{i_1, \dots, i_k\} \in \mathbf{I}_{\text{alf}}$, que representam os efeitos do ambiente, o processo de evolução populacional de uma geração para outra pode ser esquematizado como mostra a figura 2.2.

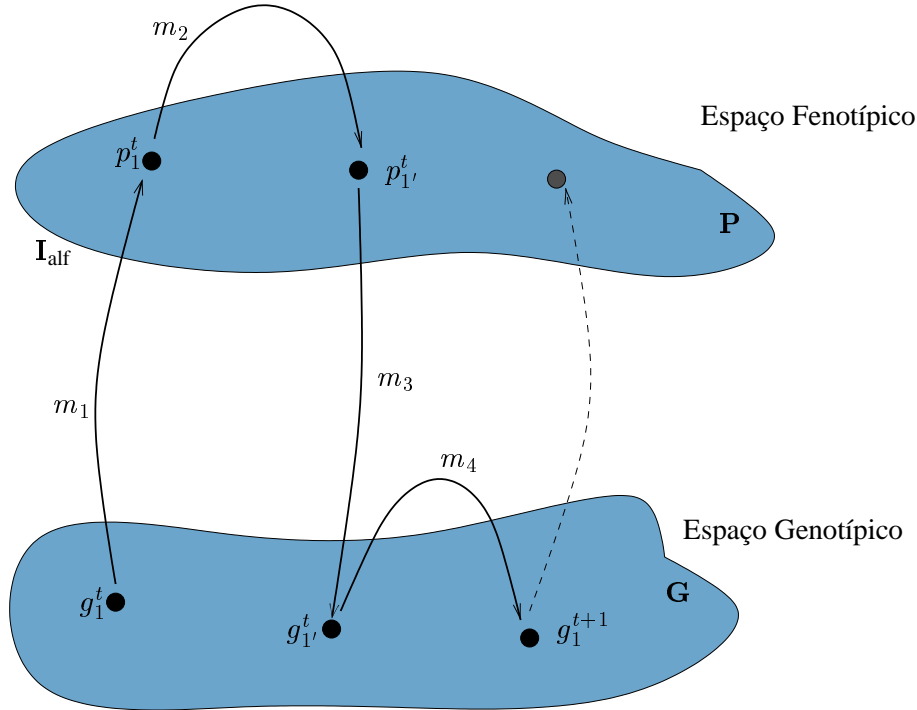


Figura 2.2: Evolução da população durante uma geração (Genótipo \times Fenótipo).

Neste caso, existem 4 mapeamentos atuando no processo:

$$\begin{aligned} m_1 &: \mathbf{I}_{\text{alf}} \times \mathbf{G} \rightarrow \mathbf{P}, \\ m_2 &: \mathbf{P} \rightarrow \mathbf{P}, \\ m_3 &: \mathbf{P} \rightarrow \mathbf{G}, \\ m_4 &: \mathbf{G} \rightarrow \mathbf{G}. \end{aligned}$$

O mapeamento m_1 , denominado epigênese, mapeia elementos g_1^t , ou seja, as codificações genéticas pertencentes a \mathbf{G} na geração t , em uma coleção particular de fenótipos $p_1^t \in \mathbf{P}$, os quais são também função dos símbolos de entrada produzidos pelo ambiente \mathbf{I}_{alf} . Este mapeamento é do tipo muitos-para-um, pois existe uma infinidade de genótipos que podem resultar num mesmo fenótipo (Von Zuben, 2000).

O mapeamento m_2 , ou seleção, mapeia o conjunto p_1^t em $p_{1'}^t$. Este mapeamento descreve os processos de seleção e migração de indivíduos dentro da população local. Como a seleção natural

opera apenas nas expressões fenotípicas do genótipo, o código g_1^t não está envolvido no mapeamento m_2 . Sob um processo de competição por recursos limitados e seleção natural, indivíduos mais bem adaptados têm mais probabilidade de sobreviver e propagar seu material genético (Atmar, 1994).

O mapeamento m_3 , conhecido como representação ou sobrevivência genotípica, descreve os efeitos sobre o genótipo dos processos de seleção e migração realizados pelo mapeamento m_2 . Estes efeitos, medidos no espaço genotípico \mathbf{G} , são avaliados pelo número de sobreviventes p_1^t , que são representados pelos genótipos g_1^t .

Finalmente, no mapeamento m_4 , a mutação e o crossover mapeiam os códigos $g_1^t \in \mathbf{G}$ em g_1^{t+1} também em \mathbf{G} . Este mapeamento descreve as regras de mutação e crossover e abrange todas as alterações genéticas. Com a criação da nova população de genótipos g_1^{t+1} , uma geração está completa e uma nova seqüência de mapeamentos se inicia.

A adaptação evolutiva ocorre em sucessivas iterações destes mapeamentos. Uma medida singular desta adaptação evolutiva é dada pela habilidade de uma população em antecipar a seqüência de símbolos (elementos do conjunto \mathbf{I}_{alf}) que irão provir do ambiente. Sob este ponto de vista, a ausência de adaptabilidade é medida pelos custos e conseqüências de previsões incorretas (por parte dos fenótipos codificados na população de genótipos) da ocorrência seqüencial de símbolos ambientais. Neste sentido, podemos dizer que a evolução minimiza a surpresa. No contexto da abordagem que está sendo proposta, \mathbf{I}_{alf} poderia representar os dados de treinamento e teste. Portanto, a presença da adaptabilidade estaria associada à capacidade dos sistemas nebulosos (fenótipos) de responderem corretamente a dados utilizados na fase de treinamento e a minimização da surpresa estaria relacionada à capacidade dos sistemas de responderem de maneira satisfatória a dados não utilizados na fase de treinamento (dados de teste), ou seja, sistemas com alto grau de adaptação evolutiva representam modelos nebulosos acurados e com boa capacidade de generalização.

No processo evolutivo, uma população de genótipos é mapeada em seus respectivos fenótipos que por sua vez são mapeados na superfície de desempenho (*fitness surface*). A figura 2.3 ilustra este conceito.

A superfície de desempenho mapeia todo elemento de \mathbf{P} em um valor real não-negativo (*fitness*). De geração em geração, é esperado que o grau médio de adaptação da população aumente. Se p_A^t representa o conjunto de fenótipos de uma população A (e não apenas um elemento de \mathbf{P}), e p_B^t de uma população B , então $\overline{F}(p_A^t)$ e $\overline{F}(p_B^t)$ representam o grau médio de adaptação de cada população. Na próxima geração, é esperado que $\overline{F}(p_A^{t+1}) > \overline{F}(p_A^t)$ e $\overline{F}(p_B^{t+1}) > \overline{F}(p_B^t)$, embora esta condição possa ser violada esporadicamente sem prejudicar a evolução. Isto se deve ao fato de

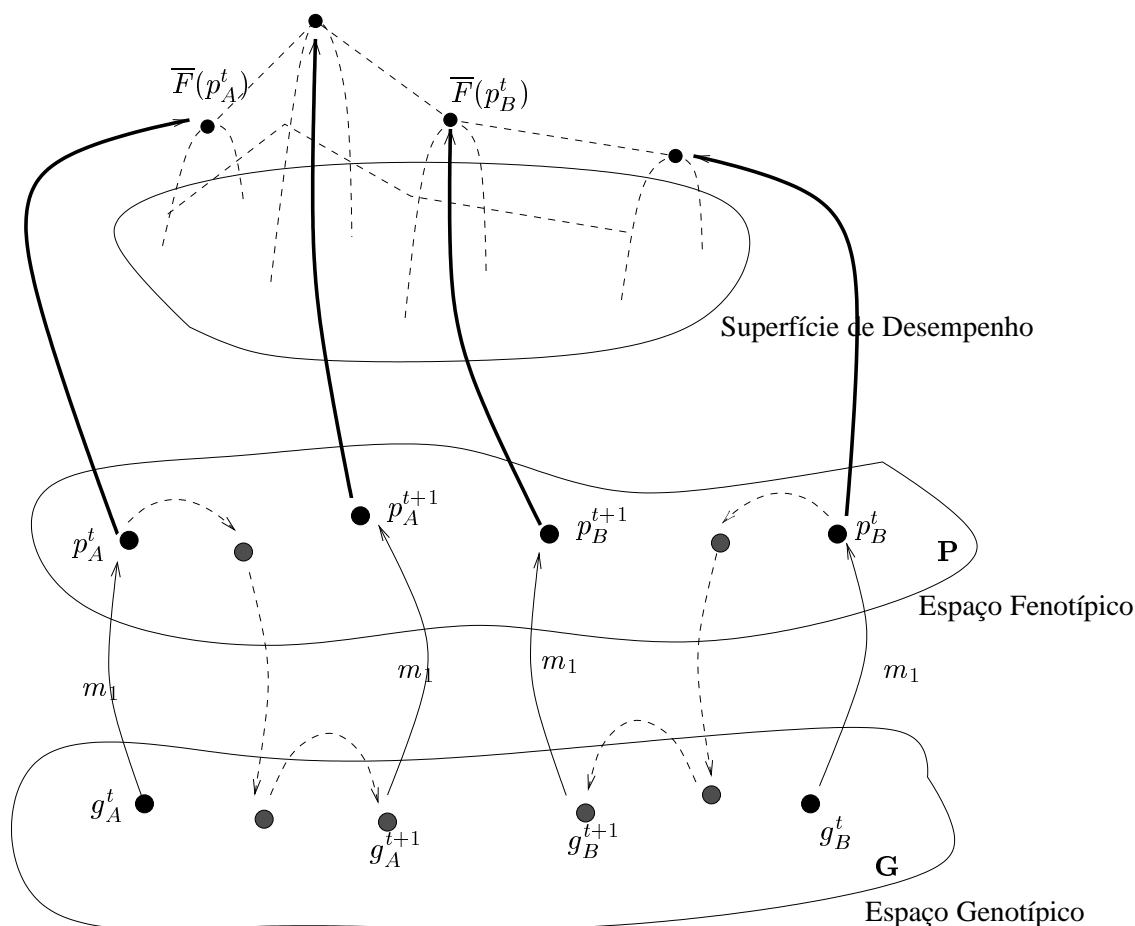


Figura 2.3: Avaliação do desempenho médio de uma população de uma geração para outra.

que os melhores indivíduos de cada população podem ter o seu grau de adaptação aumentado sem que a média da população aumente. Sendo assim, o fundamental é que exemplares da população tendam aos picos da superfície de desempenho, ou seja, explorem regiões promissoras do espaço de busca.

Existem pesquisas que propõem uma visão invertida da superfície de adaptação: populações avançam descendo picos da superfície de adaptação até que um ponto de mínimo seja encontrado. Qualquer que seja o ponto de vista, a evolução é inerentemente um processo de otimização. A seleção produz fenótipos tão próximos do ótimo quanto possível, dadas as condições iniciais, restrições ambientais e parâmetros evolutivos. Entretanto, em sistemas biológicos reais e problemas de aplicação com objetivos e condições variantes no tempo, não existem superfícies de adaptação estáticas. O ambiente está em constante mudança, fazendo com que populações estejam em constante evolução em direção a novas regiões promissoras (Wineberg, 2000).

2.3 Algoritmos Genéticos

O algoritmo genético clássico, idealizado por Holland (1975), permite partir de uma população inicial e produzir novas gerações de indivíduos, utilizando um método de seleção e operadores de mutação e crossover. Na notação clássica, cada indivíduo da população é representado por um arranjo de elementos que podem assumir um dentre dois valores possíveis (lista de atributos do tipo binário). Em outras palavras, o indivíduo é representado por um vetor (cromossomo) no qual cada componente (gene), em uma posição específica (*locus*), pode assumir um dentre dois valores (alelos). O AG clássico serviu como fonte de inspiração para uma série de AGs que vieram posteriormente. Mas a principal contribuição de Holland foi a base para a análise teórica dos algoritmos genéticos (*Schema Theory*) que foi posteriormente estendida para diferentes tipos de representação e operadores genéticos (Holland, 1975; Vose, 1991).

Um fator importante a ser analisado trata do espectro de aplicação dos algoritmos genéticos. Baseado nas possibilidades de aplicação e uso de conhecimento específico do problema, estratégias de solução de problemas podem ser divididas em métodos *fracos* e *fortes* (Michalewicz, 1996). Um método fraco tem universo de aplicação mais amplo, uma vez que supõe pouco conhecimento específico disponível. Entretanto, este pode sofrer de explosão combinatória quando utilizado para solucionar problemas em larga escala. Isto pode ser evitado incorporando-se conhecimento específico à metodologia utilizada para solucionar estes problemas (os chamados métodos fortes), com a desvantagem da redução do universo de aplicação. Os algoritmos genéticos clássicos, reconhecidos como métodos fracos, são ferramentas de propósito geral, podendo ser usados em uma série de problemas práticos, sem modificações em sua estrutura básica. No entanto, não existe um consenso na opinião de quando o uso de AG é apropriado e quando o uso de técnicas alternativas (métodos fortes) é mais indicado. Segundo abordado por Mitchell (1996), algumas características podem ajudar neste processo de decisão. A seguir, algumas situações favoráveis ao uso de AG (em relação a outros métodos de busca) são listadas:

- O espaço de busca é grande.
- A superfície de desempenho é mal comportada, mas apresenta comportamento modal.
- Não existem maiores informações sobre a superfície de desempenho.

Se o espaço de busca não é suficientemente grande, este pode ser exaustivamente explorado, implicando na certeza de se encontrar a melhor solução. Nestes casos, o uso de AG é pouco indicado pois pode levar à convergência para um ótimo local ao invés do ótimo global. No caso de

superfícies de desempenho unimodais e suaves (bem comportadas), algoritmos baseados no método do gradiente (*hill-climbing*, por exemplo), podem ser mais apropriados. Estes métodos são mais hábeis no aproveitamento da suavidade, especificando a direção de busca de forma determinística a cada geração. Se características específicas do problema são bem conhecidas (como no caso do problema do caixeiro viajante), métodos de busca que incorporam heurísticas dedicadas podem ter desempenho melhor quando comparados aos métodos de propósito geral, como os AGs. Na presença de pequenas quantidades de ruído na avaliação da função de desempenho, métodos que utilizam um único candidato à solução podem ser mais apropriados, pois permitem a implementação de processos de filtragem que demandam grande quantidade de recursos computacionais. Nestes casos, os AGs que operam acumulando estatísticas de desempenho de toda a população ao longo de muitas gerações podem ter custo computacional exagerado, tornando o problema intratável do ponto de vista computacional.

Estas observações são intuitivas e levam em conta a característica de dispositivos de propósito geral, considerada presente nos AGs clássicos. É óbvio que a incorporação de heurísticas dedicadas pode fazer com que situações antes desfavoráveis se tornem apropriadas para o uso dos AGs.

Além das características do espaço de busca, o desempenho dos AGs depende ainda de uma série de outros aspectos relevantes e interligados. Conforme apresentado por Iyoda (2000), a aplicação de um algoritmo genético na solução de um problema específico requer a definição apropriada dos seguintes elementos:

- codificação;
- população inicial;
- função de desempenho;
- método de seleção;
- operadores genéticos: crossover e mutação.

A codificação envolve a representação genética para candidatos à solução. A escolha da estrutura de dados a ser utilizada na codificação representa uma importante etapa na solução de problemas via algoritmos genéticos. A definição da população inicial P^0 , incluindo suas dimensões, depende do tipo de codificação adotada e, normalmente, os valores iniciais são obtidos de forma aleatória. Algum conhecimento disponível pode ser utilizado na geração da população inicial para, por exemplo, gerar somente indivíduos válidos (em problemas envolvendo restrições), ou para gerar alguns

indivíduos com características que os aproximem da solução desejada. A definição da função de desempenho F depende do problema a ser solucionado. Esta função associa, a cada indivíduo da população (solução potencial), um valor de *fitness* que mede a sua capacidade de resolver o problema. A determinação dos operadores do processo evolutivo envolve não só a definição do método de seleção a ser adotado, mas também tipos e taxas dos operadores genéticos (crossover e mutação) utilizados.

O esquema ilustrado na Figura 2.4 mostra as etapas básicas de um AG ao longo do processo evolutivo.

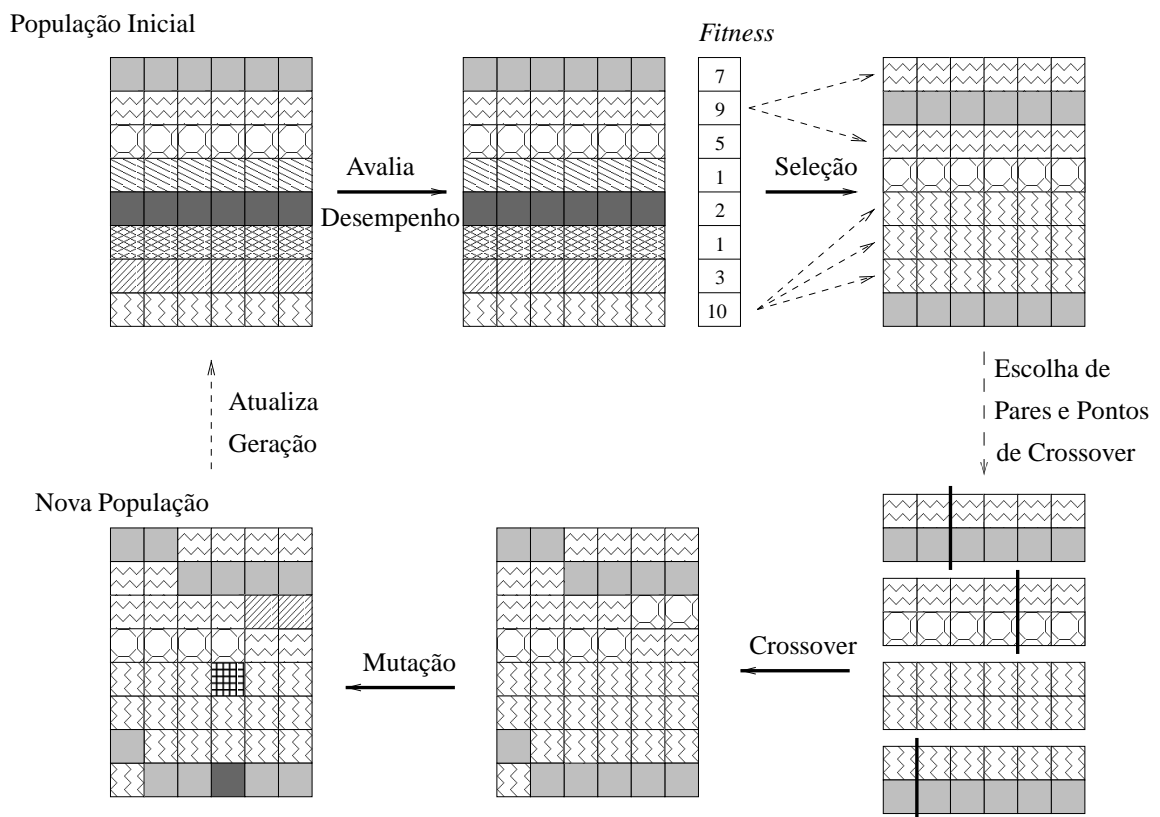


Figura 2.4: Esquema geral de evolução de um algoritmo genético.

As três decisões mais importantes, que incluem a escolha da codificação, definição do método de seleção e fixação dos operadores genéticos, são fundamentais no desempenho do AG. As seções a seguir discutem os tipos mais comumente encontrados na literatura.

2.3.1 Codificação

A codificação (genótipo) de cada indivíduo da população, que representa um candidato à solução do problema (fenótipo), é representada pela estrutura de dados que implementa o cromossomo.

Embora o AG clássico (que emprega codificação binária) tenha sido a fonte de inspiração para todas as abordagens que se seguiram, a codificação binária além de restringir o universo de aplicação, pode levar a um desempenho ruim em problemas de otimização numérica, como atesta Michalewicz (1996). As propostas mais recentes adotam populações de indivíduos representados por estruturas de dados mais genéricas, nas quais cada componente pode assumir valores pertencentes a um alfabeto qualquer.

Um exemplo das variedades de representações possíveis é a programação genética (PG), proposta por Koza (1992b), que pode ser vista como uma extensão dos algoritmos genéticos. A codificação em forma de árvore é a principal característica dos algoritmos baseados em programação genética. A maior contribuição do grupo de pesquisadores de PG está documentada em Koza (1992b,a) e Koza *et al.* (1998), demonstrando que esta técnica pode ser aplicada para produzir, de forma automática e eficiente, programas de computadores. A área de aplicação da PG é extensa e vai desde a regressão simbólica até a evolução de circuitos elétricos analógicos.

A despeito das diferentes estruturas de codificação possíveis, o cromossomo será aqui representado como um vetor (h_i^t) de tamanho fixo s na forma:

$$h_i^t = \langle v_{i1}^t, v_{i2}^t, \dots, v_{is}^t \rangle, \text{ onde } v_{ik}^t \in \text{Alf}_k, \text{ para } 1 \leq k \leq s,$$

onde Alf_k representa o conjunto de valores possíveis (alfabeto) que podem ser atribuídos ao k -ésimo elemento do vetor h_i^t . Este cromossomo pode ser interpretado como uma seqüência de atributos (genes), do i -ésimo indivíduo da população, sendo que cada gene no *locus* k é representado, na geração t , por seu alelo v_{ik}^t . Esta forma mais geral de representação supõe que cada gene (no *locus* k) tem seu próprio alfabeto Alf_k (finito ou infinito). Cria-se, assim, a possibilidade de representação de seqüências de estruturas mais complexas, por exemplo, combinando codificação inteira e real num mesmo cromossomo (codificação mista). Neste caso, a implementação mista estabeleceria dois tipos de codificação possíveis para cada *locus* k :

Codificação Inteira: $v_{ik}^t \in \text{Alf}_k = \{a_1^k, a_2^k, \dots, a_{\text{lim}}^k\}$, onde $a_j^k \in \mathcal{Z}$, $1 \leq j \leq \text{lim}$;

Codificação Real: $v_{ik}^t \in D_k = [l_k, u_k] \subset \mathfrak{R}$,

onde l_k e u_k indicam os limites inferior e superior do intervalo D_k .

Um questão importante no processo de codificação está ligada às restrições impostas às soluções candidatas. Estas restrições correspondem à inclusão de conhecimento específico do problema na codificação da solução, resultando em AGs mais dedicados. Segundo Davis (1987), restrições que não devem ser violadas podem ser implementadas de três formas distintas: impondo-se altas penalidades aos indivíduos que violam estas restrições; adotando-se penalidades moderadas; ou através de decodificadores com etapas de reparação, que evitam a criação de soluções inactíveis. Cada uma das três formas apresenta suas vantagens e desvantagens. A adoção de penalidades altas ou moderadas corre o risco de levar à criação de AGs que despendem um tempo elevado na avaliação de indivíduos inactíveis. Além disso, no caso de penalidades altas, a convergência para soluções que atendam apenas o critério de factibilidade, ao invés de soluções que também considerem o nível de desempenho, pode ocorrer com maior frequência. Isto porque a obtenção de boas soluções pode depender da produção de soluções intermediárias inactíveis que são eliminadas por altas penalidades na função de desempenho. Por outro lado, o critério que adota uso de penalidades moderadas pode resultar em soluções que violem parte das restrições, pois o benefício de aceitar estas soluções (parcialmente factíveis) pode ser maior do que o de rejeitá-las. No caso da utilização de decodificadores que evitam a construção de indivíduos ilegais, além do custo computacional ser normalmente maior, a implementação de todas as etapas de reparação nem sempre é possível.

Embora importante, a codificação não é o único fator que garante a construção de soluções factíveis. Qualquer problema que envolva restrição requer uma análise cuidadosa que resulte não só na adoção de uma representação adequada mas também na escolha de operadores genéticos apropriados. A abordagem a ser adotada neste trabalho garante a construção de soluções sempre factíveis, tanto pela escolha da codificação apropriada, como pela utilização de operadores genéticos com etapas de reparação.

2.3.2 Métodos de Seleção

Após decidir a codificação, o passo seguinte trata da definição do método de seleção a ser utilizado, o qual envolve a escolha dos indivíduos na população que irão criar descendentes e quantos descendentes serão criados. O objetivo da seleção é privilegiar, em média, os indivíduos melhor adaptados, na esperança de que seus descendentes tenham desempenho ainda melhor. A seleção deve ser tal que produza um balanço adequado entre a pressão seletiva e a variação introduzida pelos operadores genéticos (Mitchell, 1996). Métodos de seleção muito fortes tendem a gerar super-indivíduos

(indivíduos com medida de desempenho muito superior aos demais), reduzindo a diversidade necessária para alterações e progressos futuros. A geração de super-indivíduos pode levar a uma convergência prematura do processo evolutivo. Por outro lado, métodos de seleção excessivamente fracos (pouca pressão seletiva) tendem a produzir progressos muito lentos na evolução. A seguir, serão descritos alguns dos métodos mais comumente encontrados na literatura.

Seleção Proporcional ao Desempenho

Neste método, a probabilidade de um indivíduo se reproduzir na próxima geração é diretamente proporcional à sua medida de desempenho (*fitness*). Para uma população $P^t = \{h_1^t, h_2^t, \dots, h_{S_p}^t\}$, na qual o i -ésimo indivíduo h_i^t tem associado a ele uma medida de desempenho positiva e não nula ($F(h_i^t) \in \mathfrak{R}_0^+$), a probabilidade Pr_{sel}^i deste indivíduo ser selecionado é dada por:

$$Pr_{\text{sel}}^i = \frac{F(h_i^t)}{\sum_{j=1}^S F(h_j^t)}.$$

Este esquema é também conhecido como seleção por roleta (*roulette wheel*), pois o *fitness* pode ser representado como segmentos consecutivos em uma roleta imaginária e a probabilidade de escolha de um segmento depende diretamente de seu tamanho. A figura 2.5, ilustra como seria a roleta para os indivíduos representados na geração t , mostrada na figura 2.4.

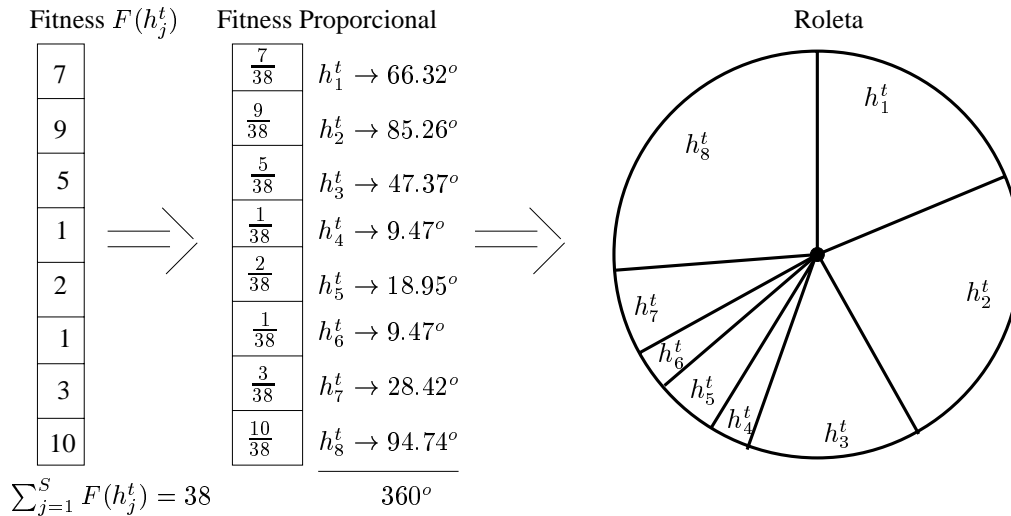


Figura 2.5: Exemplo de roleta na seleção proporcional ao desempenho.

A seleção proporcional ao desempenho pode ser problemática se os indivíduos da população apresentarem desempenhos muito próximos entre si. Além disso, se o tamanho da população é

pequeno, a perda de diversidade pode levar à convergência prematura, pois a busca fica reduzida a poucos pontos, causando uma diminuição no poder de exploração do AG.

Seleção Baseada na Classificação (*Rank*)

Uma opção para se evitar o surgimento de super-indivíduos (relativos aos demais existentes na população atual) e a ocorrência de convergência prematura é reduzir as diferenças entre estes, supondo uma classificação ρ para cada indivíduo h_i^t , ao invés de pegar diretamente sua medida de desempenho (*fitness*) $F(h_i^t)$.

Os indivíduos da população $P^t = \{h_1^t, h_2^t, \dots, h_{S_p}^t\}$, são arranjados em ordem crescente de desempenho, tal que $F(h_i^t) \leq F(h_j^t)$ para todo $1 \leq i \leq j \leq S_p$. Seja $\rho(h_i^t)$ a classificação do indivíduo $h_i^t \in P^t$, tal que o seu desempenho seja dado por $F_{rank}(h_i^t) = S_p - \rho(h_i^t) + 1$. Então, a probabilidade de seleção deste indivíduo é dada por:

$$Pr_{sel}^i = q - (\rho(h_i^t) - 1)p. \quad (2.1)$$

Como $\sum_{i=1}^{S_p} Pr_{sel}^i = 1$ tem-se

$$q = p \frac{S_p - 1}{2} + \frac{1}{S_p}.$$

Portanto, a equação 2.1 pode ser reescrita como:

$$Pr_{sel}^i = q - \frac{2(q - 1/S_p)}{(S_p - 1)}(\rho(h_i^t) - 1),$$

onde q define o nível de pressão seletiva. Este esquema permite ao usuário estabelecer o grau de pressão seletiva que deseja impor ao processo evolutivo, definindo $q = \frac{1}{S_p}$ para pressão seletiva mínima e $q = \frac{2}{S_p}$ para pressão seletiva máxima (Michalewicz, 1996).

A seleção por classificação tem a desvantagem de exigir a ordenação de toda a população, o que pode representar um custo computacional excessivo em algumas aplicações específicas.

Seleção por Torneio

A seleção por torneio é similar à seleção por classificação em termos de pressão seletiva, mas pode ser mais eficiente do ponto de vista computacional, principalmente em implementações paralelas. Seja r uma realização da variável aleatória χ com distribuição uniforme no intervalo $[0, 1]$, definida daqui por diante como $\chi \sim \mathcal{U}[0, 1]$ e c um parâmetro qualquer definido no intervalo unitário.

As regras da seleção por torneio são as seguintes:

1. escolher aleatoriamente dois indivíduos h_i^t e h_j^t (suponha $F(h_i^t) \geq F(h_j^t)$) como possíveis candidatos para a nova população P^{t+1} ;
2. Obter r ;
3. escolher o melhor indivíduo (h_i^t) se $r > c$; e o pior caso contrário.
4. os dois indivíduos voltam à população original P^t ;
5. retorna ao passo 1.

Existem variações deste algoritmo nas quais a escolha dentre os dois candidatos é determinística (escolhe-se sempre o melhor). Há outras abordagens nas quais o torneio é realizado, não por dois, mas por uma porcentagem da população, e o melhor indivíduo deste grupo é selecionado. Este último tipo de seleção (melhor do grupo vence) será aquele adotado nas simulações ilustradas no capítulo 5.

Elitismo

O termo elitismo, introduzido por De Jong (1975), está associado à adoção de uma operação adicional junto aos métodos de seleção, que força o AG a reter o melhor indivíduo ou um número de melhores indivíduos, a cada geração. Estes indivíduos poderiam ser perdidos se não fossem selecionados de forma determinística para compor a próxima geração, ou então fossem modificados por crossover ou mutação. Em grande parte dos casos, estratégias elitistas associadas aos métodos de seleção melhoram o desempenho do AG (Mitchell, 1996).

Outros Métodos

Há vários outros métodos (determinísticos ou não) propostos para a implementação do processo de seleção, como por exemplo a seleção puramente aleatória, na qual todos os indivíduos têm a mesma probabilidade de serem escolhidos. Normalmente, este método é utilizado em conjunto com estratégias elitistas. Outro exemplo é o método de seleção *steady-state*, que mantém a população original de uma geração para outra, com exceção de poucos indivíduos que são substituídos por descendentes do melhor, obtidos por mutação ou crossover. Neste método, a escolha da população inicial tem papel muito importante, pois a evolução é mais lenta e obtida com pequenas modificações a cada geração. Na direção contrária, aparece a seleção por diversidade, em que parte da população é obtida escolhendo-se os indivíduos mais diversos, a partir do melhor indivíduo, à custa

do uso de recursos computacionais adicionais. O conceito de mais diverso é baseado em algum critério de distância previamente definido.

2.3.3 Operadores Genéticos

Após a escolha da codificação e método de seleção, o passo final para a aplicação do AG trata do processo reprodutivo. A reprodução, que envolve a aplicação dos operadores genéticos junto aos indivíduos selecionados, é responsável pela obtenção das novas soluções. A variação genética, principal ingrediente na evolução, é quem garante a exploração de novas estruturas genotípicas e suas funcionalidades (retratadas nos fenótipos). Os operadores genéticos representam a fonte de diversidade e variabilidade.

A escolha dos operadores genéticos está intimamente ligada à codificação adotada para a representação genética. Os operadores genéticos mais comuns são o crossover, também conhecido como recombinação, e a mutação. A mutação introduz novos alelos e pode produzir combinações genéticas até então inéditas na codificação genotípica. Na mutação, informações que foram perdidas ao longo do processo evolutivo podem ser resgatadas, ou mesmo informações totalmente novas podem ser introduzidas (ver figura 2.4). O crossover, ao combinar os indivíduos trocando suas informações genéticas, também traz alguma inovação. Mas no processo de crossover há uma tendência maior de gerar cromossomos com combinações de características já existentes na população, resultando em um efeito mais visível de convergência (Bäck *et al.*, 2000a,b).

Crossover

A operação de crossover troca parte da informação genética entre dois (ou mais) indivíduos, produzindo novas soluções potenciais com algumas características dos pais. Os efeitos da recombinação de códigos genéticos são mais previsíveis sempre que pleiotropia e poligenia não estiverem presentes. Conforme ilustrado na figura 2.4, o processo de crossover depende da escolha dos pares (ou grupos) de indivíduos, que pode ser feita de várias formas distintas, dentre as quais se destacam:

- os dois (ou mais) indivíduos são escolhidos de forma aleatória;
- um indivíduo é escolhido com base no desempenho e o outro (ou o restante) é escolhido de forma aleatória.

Outros fatores importantes são a taxa de crossover e o(s) ponto(s) de troca de informação. A taxa define com que probabilidade (P_C) os indivíduos da população sofrerão a recombinação.

O ponto de troca da informação depende do tipo de crossover a ser adotado. Existem três tipos mais comuns: crossover simples (também conhecido como crossover de 1-ponto), crossover de múltiplos pontos e crossover uniforme. A figura 2.6 mostra as características de cada tipo, supondo a escolha prévia de dois indivíduos $h_i^t = \langle v_1^t, v_2^t, \dots, v_s^t \rangle$ e $h_j^t = \langle u_1^t, u_2^t, \dots, u_s^t \rangle$, $s = 10$, para participarem do processo de crossover.

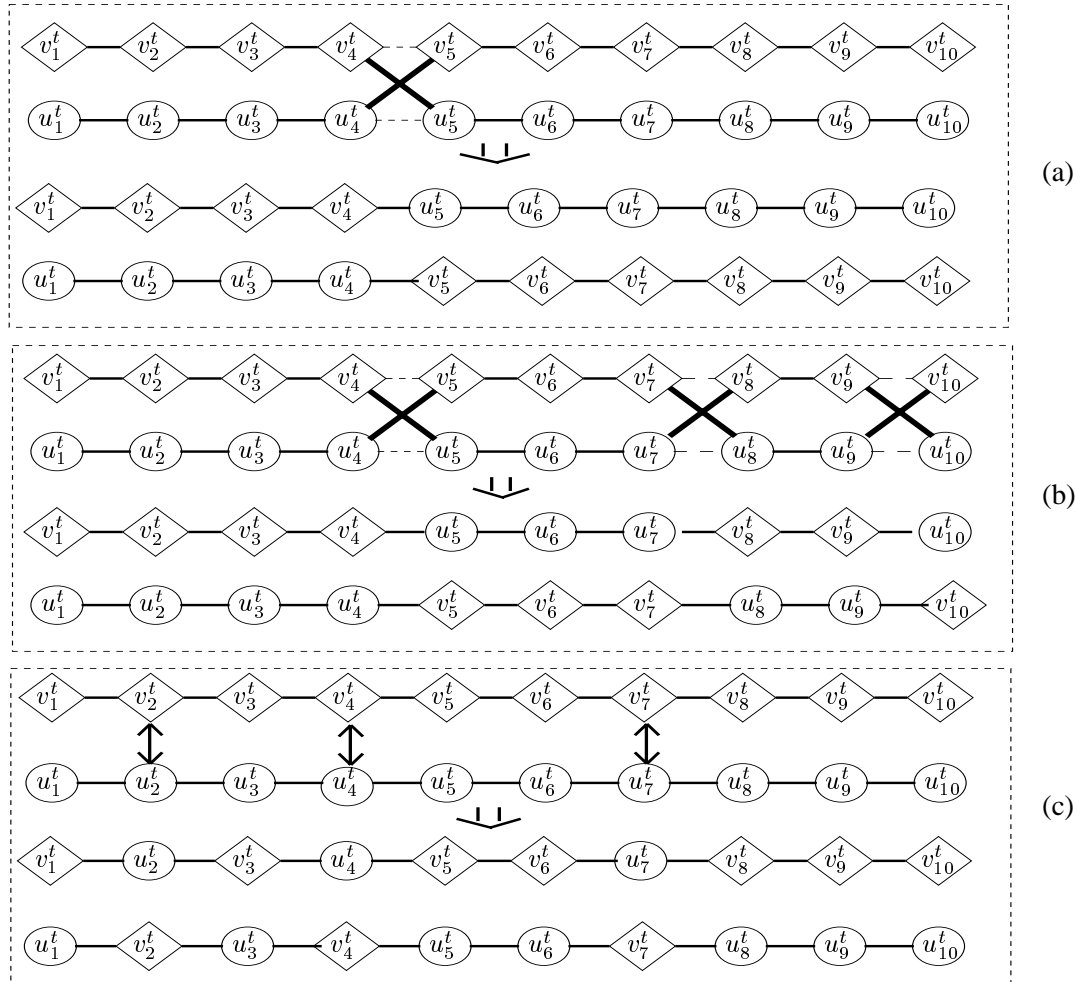


Figura 2.6: Operadores de crossover: a) crossover simples; b) Crossover de múltiplos pontos; c) Crossover uniforme.

No crossover simples, a posição de troca, também conhecida como ponto de corte, é escolhida de maneira aleatória. Este tipo de operador impõe a quebra da seqüência genética, definindo qual a proporção de informação dos pais que cada descendente receberá. Uma característica do crossover simples é a maior probabilidade do ponto de corte cair longe dos extremos do cromossomo. Como alternativa a essa possível concentração nos *loci* centrais, foi proposto o operador de crossover de

múltiplos pontos, o qual escolhe de forma aleatória mais de uma posição de troca. Uma terceira alternativa desconsidera a relevância da ordenação dos genes. Este operador, conhecido como crossover uniforme, determina a porcentagem de alelos que serão trocados e escolhe aleatoriamente as posições que serão permutadas.

Estes três tipos de crossover foram concebidos para uso com codificação binária. Entretanto, não existe nenhum impedimento quanto ao uso destes operadores junto a outros tipos de codificação. Mas, para problemas com restrições, existem operadores de crossover específicos, que garantem a factibilidade dos descendentes originários de indivíduos factíveis. Um exemplo é a proposta de crossover aritmético apresentada por Michalewicz (1996), para o caso de regiões factíveis convexas. Neste tipo de operador, os descendentes são obtidos como combinação convexa dos pais, garantindo-se assim a factibilidade das soluções resultantes.

Algumas abordagens para problemas com restrições apresentadas na literatura sugerem o uso de algoritmos reparadores associados aos operadores genéticos (Bäck *et al.*, 2000a). Estes algoritmos modificam a codificação genética resultante do crossover ou da mutação de forma que as novas soluções obtidas não violem as restrições impostas às soluções do problema.

Mutação

A mutação altera um ou mais genes de um cromossomo. A taxa de mutação define com que probabilidade (P_M) será modificado cada alelo presente no cromossomo. A mutação ocorre normalmente a taxas reduzidas, pois a idéia do operador de mutação é incorporar novas codificações genéticas, sem no entanto destruir o progresso obtido no processo evolutivo. A variabilidade obtida pela mutação deve ter o comportamento de uma perturbação de efeito localizado (Iyoda, 2000). A aplicação do operador de mutação depende da escolha do ponto de mutação ou posição sorteada. Normalmente esta escolha é aleatória (todos os pontos têm a mesma probabilidade de serem escolhidos). No entanto, conforme será visto no capítulo 4, a abordagem proposta permite um esquema alternativo de escolha do ponto de mutação que é baseado no *fitness*.

O processo de mutação é implementado com base em um operador unário que transforma v_k^t em $v'_k{}^t$. Existem três tipos mais comuns de mutação: simples, inteira e real. Na mutação simples (usada na codificação binária), troca-se o valor do bit. Na mutação inteira, a busca é feita sobre um alfabeto de valores inteiros $\text{Alf}_k \subset \mathcal{Z}$. Na mutação real, o operador pode realizar uma busca ampla sobre um domínio real ao longo de todo o processo evolutivo, como é o caso da mutação uniforme, ou realizar uma busca mais abrangente no início da evolução e mais localizada nas gerações finais, como no caso da mutação não-uniforme (Michalewicz, 1996).

Seja r_1 uma realização da variável aleatória $\chi_1 \sim \mathcal{U}[0, 1]$.

- Mutaç o simples: O valor antigo v_k^t na posiç o sorteada k   substituído pelo novo alelo $v_k'^t$ que representa o seu complemento.

$$v_k'^t = \begin{cases} 1 & \text{se } r_1 \leq Pr_{mut} \text{ e } v_k^t = 0, \\ 0 & \text{se } r_1 \leq Pr_{mut} \text{ e } v_k^t = 1, \\ v_k^t & \text{se } r_1 > Pr_{mut}. \end{cases}$$

- Mutaç o inteira: O novo alelo $v_k'^t$   elemento do conjunto $\text{Alf}_k \subset \mathcal{Z}$.

$$v_k'^t = \begin{cases} a_k \in \text{Alf}_k - \{v_k^t\} & \text{se } r_1 \leq Pr_{mut}; \\ v_k^t & \text{se } r_1 > Pr_{mut}. \end{cases}$$

- Mutaç o real: o novo alelo $v_k'^t$   um valor real pertencente ao dom nio $D_k = [l_k, u_k]$. Existem dois tipos mais comuns:

- mutaç o uniforme: o novo alelo   um valor aleat rio u (com distribuic o de probabilidade uniforme) do dom nio D_k .

$$v_k'^t = \begin{cases} u \in D_k = [l_k, u_k] & \text{se } r_1 \leq Pr_{mut} \\ v_k^t & \text{se } r_1 > Pr_{mut}, \end{cases}$$

- mutaç o n o-uniforme: o novo alelo   obtido pela adiç o (ou subtraç o) de uma funç o Δ que depende dos limites do Dom nio $D_k = [l_k, u_k]$ e da geraç o t .

$$v_k'^t = \begin{cases} v_k^t + \Delta(t, u_k - v_k^t) & \text{se } r_1 \leq Pr_{mut} \text{ e } r_2 \leq 0.5, \\ v_k^t - \Delta(t, v_k^t - l_k) & \text{se } r_1 \leq Pr_{mut} \text{ e } r_2 > 0.5, \\ v_k^t & \text{se } r_1 > Pr_{mut}, \end{cases}$$

onde r_2   uma realizaç o de $\chi_2 \sim \mathcal{U}[0, 1]$ e a funç o $\Delta(t, z)$ retorna um valor no intervalo $[0, z]$, tal que a probabilidade de $\Delta(t, z)$ se aproximar de zero aumenta   medida que t cresce. A funç o utilizada neste trabalho   parametrizada pelo grau de n o-uniformidade b , e   dada por:

$$\Delta(t, z) = z \left(1 - r^{(1 - \frac{t}{T})^b} \right),$$

onde T   o n mero m ximo de geraç es e r   um valor aleat rio escolhido no intervalo $[0, 1]$.

A figura 2.7 ilustra exemplos de cromossomos com diferentes tipos de codificação e a aplicação de operadores de mutação simples, inteira, real uniforme e real não-uniforme.

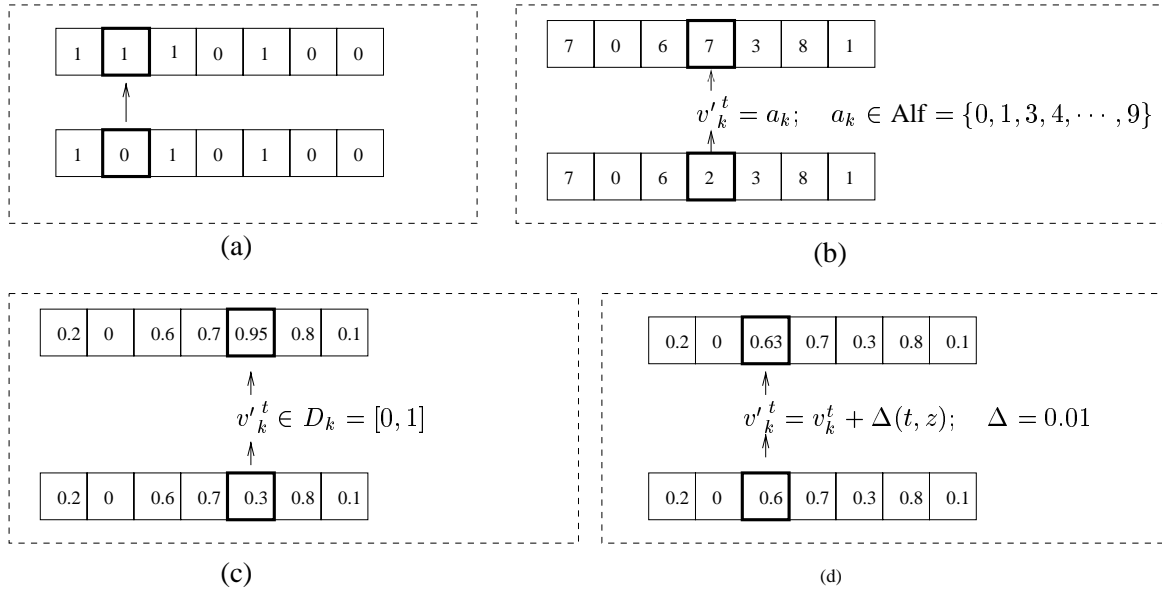


Figura 2.7: Exemplos de mutação: a) Mutação simples; (b) Mutação inteira; c) Mutação real uniforme; d) Mutação real não-uniforme.

Neste exemplo específico, a mutação simples (figura 2.7(a)) ocorre na segunda posição (*locus* $k = 2$) e troca o alelo 0 para 1. A figura 2.7(b) traz um exemplo de mutação inteira na quarta posição (*locus* $k = 4$) que altera o alelo 2 para 7. Na figura 2.7(c), a mutação real uniforme ocorre no *locus* $k = 5$ e altera o alelo para um novo valor dentro do domínio $D_k = [0, 1]$. No caso da mutação real não-uniforme ocorrida no *locus* $k = 3$ (ver figura 2.7(d)), a alteração é realizada sob um domínio estreito (supondo-se que o limite de gerações T esteja próximo de ser atingido). Verifica-se que na mutação uniforme o novo alelo independe do valor atual v_k^t e qualquer valor no intervalo especificado é igualmente provável. Por isso, algumas vezes este tipo de operação é chamado de mutação sem memória. A mutação não-uniforme, ao contrário, vai acumulando uma “história” ao longo do processo evolutivo, já que o novo valor depende sempre do valor atual e a perturbação adicionada é definida aqui como sendo inversamente proporcional ao número de gerações já transcorridas.

2.3.4 Exemplo de Aplicação

Um aspecto essencial a ser levado em conta quando se propõem soluções de problemas complexos com base em algoritmos genéticos trata da característica de propósito geral associada a esta área da computação. Outro aspecto importante é a eficiência no processo de busca propiciada pelo paralelismo inerente às buscas baseadas em AG. O objetivo desta seção é mostrar um exemplo prático do uso dos AGs, salientando estas duas características fundamentais.

O exemplo de aplicação a ser mostrado foi abordado por Von Zuben (2000) e envolve um problema de otimização para o qual não existe solução dedicada que garanta a obtenção da solução ótima, utilizando uma quantidade viável de recursos computacionais. Mesmo que soluções dedicadas mais eficazes do que aquelas obtidas por AGs possam ser encontradas, estas soluções não são facilmente extensíveis a novas situações e certamente seria necessária uma solução dedicada para cada problema.

- **Problema:** Otimização de Parâmetros de Uma Caixa Preta

Sabendo que cada botão pode ser colocado em 16 posições distintas, encontre a melhor combinação de posições para os 9 botões disponíveis na superfície da caixa preta apresentada na figura 2.8 de modo que o sinal de saída assumo o valor máximo.

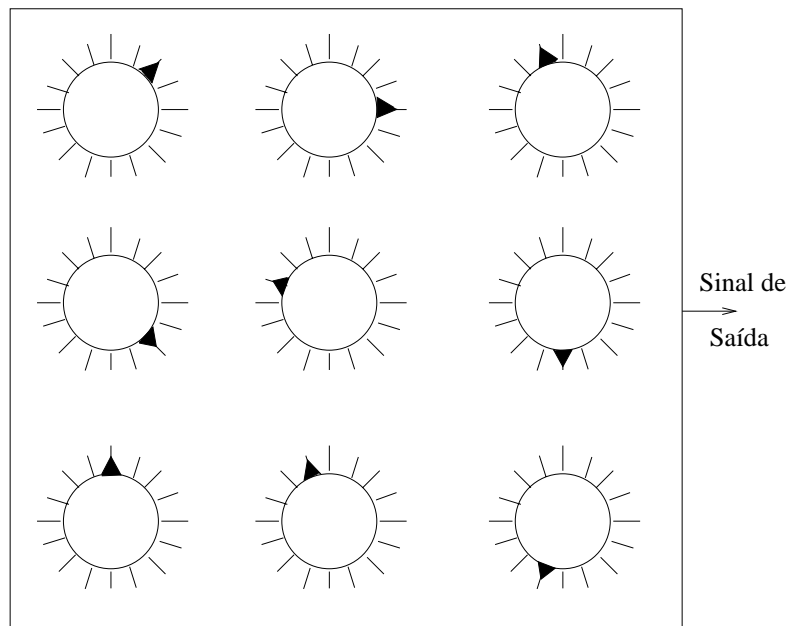


Figura 2.8: Problema de otimização da caixa preta.

- **Codificação:** Existem 16 posições possíveis para cada um dos 9 botões. Na codificação binária, 4 bits são suficientes para representar cada uma das 16 posições, conforme mostra a figura 2.9.

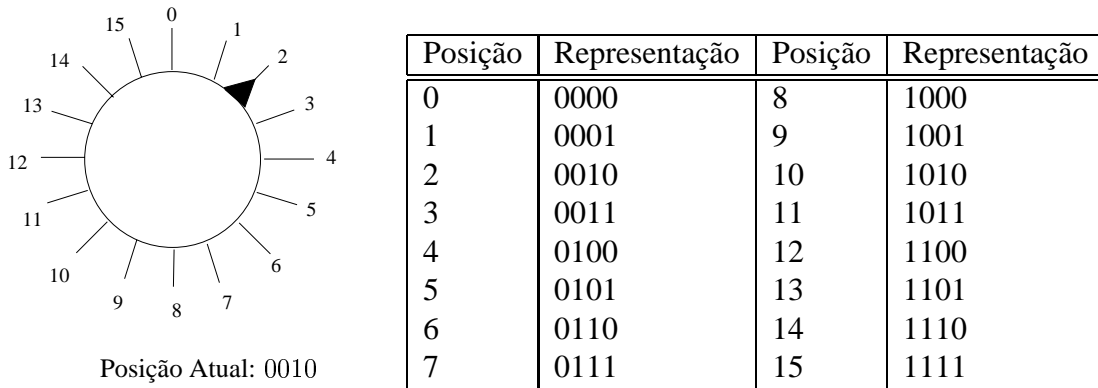


Figura 2.9: Codificação binária para o problema da caixa preta.

Baseado neste tipo de codificação, cada cromossomo associado à solução candidata do problema definido anteriormente é dado por uma seqüência de 36 bits (b_1, \dots, b_{36}) , na qual o número de possíveis configurações de botões (soluções candidatas) é $2^{36} \cong 68.72$ bilhões. Neste caso, a solução candidata, mostrada na figura 2.8, seria codificada por um cromossomo na forma:

0010 0100 1111 0110 1101 1000 0000 1111 1001 .

O mapeamento, suposto desconhecido, entre as 2^{36} posições possíveis dos botões e o sinal de saída é dado por:

$$\begin{aligned} \text{sinal de saída} = & 9 + b_2b_5 - b_{23}b_{14} + b_{24}b_4 - b_{21}b_{10} + b_{36}b_{15} - b_{11}b_{26} + b_{16}b_{17} + b_3b_{33} \\ & + b_{28}b_{19} + b_{12}b_{34} - b_{31}b_{32} - b_{22}b_{25} + b_{35}b_{27} - b_{29}b_7 + b_8b_{13} - b_6b_9 + b_{18}b_{20} - b_1b_{30} \\ & + b_{23}b_4 + b_{21}b_{15} + b_{26}b_{16} + b_{31}b_{12} + b_{25}b_{19} + b_7b_8 + b_9b_{18} + b_1b_{33} , \end{aligned}$$

onde b_k representa o bit no *locus* k do cromossomo. A denominação de caixa preta se justifica pelo fato de se ter acesso ao valor de saída para cada configuração de botões, sem no entanto se conhecer *a priori* como a configuração de botões determina o valor de saída. Cada configuração de botões representa uma solução candidata e o respectivo valor de saída será diretamente tomado como valor de adaptação ou *fitness*.

A solução ótima deste problema (também suposta desconhecida) possui um valor de *fitness* igual a 27 e é codificada por um cromossomo na forma:

1111 1011 1001 1011 1111 1011 1111 0010 1111

- Dados do Processo Evolutivo

- Tamanho da População: 1500 indivíduos.
- População Inicial: gerada aleatoriamente (a probabilidade de bits 1 nos cromossomos é 0.5).
- Método de seleção: Proporcional ao Desempenho (*Roulette Wheel*)
- Operadores Genéticos: mutação simples e crossover simples
- Taxas de Operadores (probabilidades)
 - * $P_C = 0.6$;
 - * $P_M = 0.03$.

Observação: Os parâmetros do processo evolutivo, definidos anteriormente, foram escolhidos de forma arbitrária. Uma maior eficiência poderia ser obtida com parâmetros mais cuidadosamente definidos. Entretanto, esta escolha arbitrária não compromete as conclusões a serem deduzidas.

- Resultados Obtidos: Utilizando-se os parâmetros evolutivos descritos anteriormente, a solução ótima foi encontrada na geração 57, conforme mostra o gráfico da figura 2.10.

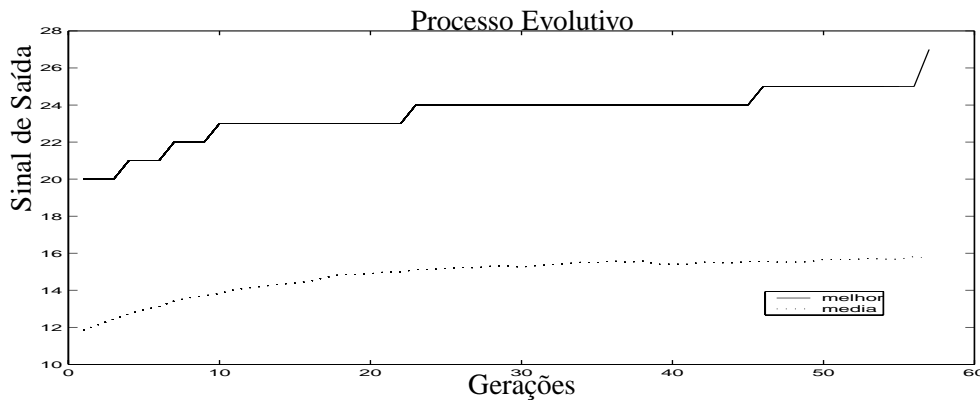


Figura 2.10: Resultado de desempenho do processo evolutivo.

Como a estratégia elitista é associada ao método de seleção baseada no desempenho, a medida de *fitness* do melhor indivíduo na população (*melhor*) nunca decresce. A medida da média de *fitness* da população (*media*), embora apresente pequenas oscilações, também mostra tendência de crescimento ao longo das gerações, reforçando a idéia de que, em média, a população como um todo melhora ao longo do processo evolutivo.

- Conclusões: O problema evidencia o grande potencial dos algoritmos genéticos como métodos de busca, já que, dentre os 68.72 bilhões de possíveis candidatos, apenas 1500 foram testados a cada geração, num total de 85500 candidatos testados, até a obtenção da solução ótima. A característica de alta flexibilidade das soluções obtidas por AG também aparece, uma vez que pequenas alterações no problema (como a inclusão de novos botões ou novas posições) seriam facilmente implementadas através de pequenas modificações na função de *fitness* e na codificação.

2.4 Co-Evolução

As propostas de solução para problemas complexos por algoritmos evolutivos podem depender da criação de paradigmas mais elaborados, para os quais a solução evolui na forma de subcomponentes, interagindo de maneira co-adaptativa. Modelos co-evolutivos parecem oferecer novas perspectivas para melhorar o desempenho de algoritmos evolutivos (Olsson, 2001). Algumas definições serão necessárias aqui:

- no jargão dos processos co-evolutivos, um nicho pode ser visto como um subespaço no ambiente, que admite diferentes tipos de comportamentos e manifestações por parte das soluções candidatas;
- uma espécie é definida como um grupo de indivíduos com características biológicas semelhantes que são capazes de se reproduzir, mas que são incapazes de gerar descendentes quando associados a indivíduos pertencentes a outros grupos ou espécies (Sareni & Krähenbühl, 1998).

O termo co-evolução foi cunhado por Ehrlich & Raven (1964) em suas descrições da provável influência que as plantas exerciam na evolução de insetos herbívoros, e vice-versa. Eles documentaram a associação entre algumas espécies de borboletas e suas plantas hospedeiras, notando que os componentes venenosos produzidos pelas plantas determinavam o uso de algumas destas plantas pelas borboletas.

Para Rosin & Belew (1997), o termo co-evolução sugere a evolução simultânea de duas ou mais espécies cuja medida de desempenho (*fitness*) esteja acoplada. Neste sentido, a co-evolução pode ser definida como uma extensão do método evolutivo tradicional.

Para Roughgarden (1976), em processos co-evolutivos, o desempenho de cada genótipo depende da densidade populacional e composição genética, não só da própria espécie, mas também das

espécies com as quais esta interage.

A utilização eficiente de algoritmos co-evolutivos depende não só da decomposição do problema em questão, mas também da definição de interdependência entre os subcomponentes, do algoritmo de recompensa para as soluções produtivas e da garantia de diversidade populacional (Potter & De Jong, 2000). Em geral, as abordagens co-evolutivas presentes na literatura podem ser classificadas como competitivas ou cooperativas.

Para algumas das abordagens competitivas, duas espécies diferentes interagem: os hospedeiros, para os quais o *fitness* é avaliado, e os parasitas, que são utilizados na avaliação dos hospedeiros. A competição acontece entre as espécies, i.e. o sucesso dos hospedeiros implica no fracasso dos parasitas (Rosin & Belew, 1997). A principal motivação para o uso de abordagens do tipo hospedeiro-parasita vem da idéia do exército armado co-evolutivo (*coevolutionary arms race*): espera-se que ao longo das gerações, cada espécie se torne mais eficiente ao explorar as deficiências da outra.

Um exemplo bem difundido de abordagem hospedeiro-parasita é a evolução de redes de ordenação utilizando AG, proposta por Hillis (1990). Neste caso, os hospedeiros representam as estratégias de ordenação e os parasitas representam os casos de teste, que são apresentados na forma de seqüência de números a serem ordenados. Especificamente, a estratégia de ordenação é avaliada baseado no modo como ela ordena os casos de teste, enquanto que os casos de teste são avaliados tendo em vista o grau de dificuldade com que são ordenados. As populações de hospedeiros e parasitas são geneticamente isoladas e a única interação acontece na função de *fitness*. Outro caso conhecido de competição é o modelo apresentado por Rosin & Belew (1997). Neste trabalho, duas espécies competitivas (hospedeiros e parasitas) são criadas e co-evoluem para solucionar problemas de aprendizado. Olsson (2001), por outro lado, explorou o papel da assimetria do problema na limitação do progresso da busca hospedeiro-parasita. Olsson (2001) mostrou que a maioria das aplicações dos algoritmos do tipo hospedeiro-parasita possui um alto grau de assimetria, e, nestes casos, as abordagens co-evolutivas tradicionais, que aplicam os operadores evolutivos nas duas populações a cada geração, podem ser menos adequadas do que aquelas que permitem que uma das espécies evolua mais rapidamente do que a outra.

Os modelos baseados em espécies competitivas são efetivos na preservação da diversidade genética e podem levar a soluções melhores quando comparados com abordagens evolutivas tradicionais. Entretanto, como apontado por Potter & De Jong (2000), uma limitação na abordagem hospedeiro-parasita reside no campo de aplicação, que é estreito devido à restrição de se decompor manualmente o problema em dois subcomponentes antagônicos.

Alguns trabalhos têm explorado o uso de populações isoladas do ponto de vista genético, representando espécies diferentes, mas que permitem a atuação cooperativa dos indivíduos destas espécies. Para muitas destas técnicas, os subcomponentes da solução emergem ao invés de serem fixados arbitrariamente. Um modelo clássico de indivíduos em processo co-evolutivo, agindo de forma cooperativa, é o sistema classificador proposto por Holland (1975). Um sistema classificador é um sistema baseado em regras (evoluídas por AG), no qual as regras individuais trabalham em cooperação para formar a solução completa do problema.

Em outro exemplo, Moriarty & Miikkulainen (1997, 1998) apresentaram uma abordagem baseada em co-evolução, com duas populações isoladas geneticamente em dois níveis distintos: uma população de neurônios e uma população de protótipos de redes neurais. A figura 2.11 mostra a cooperação das duas espécies para a formação dos modelos de redes neurais artificiais com três camadas: entrada, intermediária e saída. Cada indivíduo na população de protótipo de redes contém

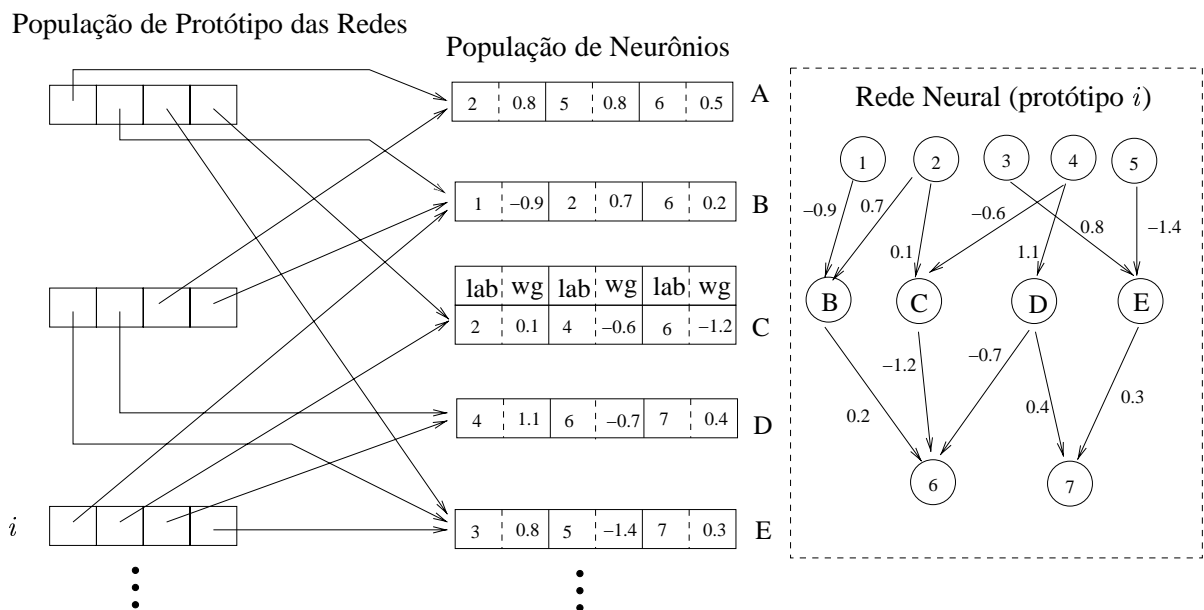


Figura 2.11: Exemplo de hierarquia na evolução de duas populações: protótipos de redes e neurônios, proposta por Moriarty & Miikkulainen (1998).

um série de apontadores (índices) para as estruturas dos neurônios. Na população de neurônios, cada cromossomo codifica um possível neurônio da camada intermediária e suas possíveis conexões (wg) com os neurônios da camada de entrada e saída da rede, identificados por seus respectivos rótulos (lab). No processo evolutivo, uma etapa fundamental é o cálculo do *fitness* que é realizado da seguinte forma:

- Escolhem-se os neurônios que farão parte de cada rede com base nos apontadores codificados no nível I (nível dos protótipos);
- Conecta-se cada rede com base nos parâmetros dos neurônios codificados no nível II (nível dos neurônios);
- Avalia-se o desempenho (*fitness*) de cada rede resultante no nível I. Este desempenho depende do domínio da aplicação;
- Avalia-se o desempenho (*fitness*) de cada neurônio codificado no nível II. Ao contrário do *fitness* dos protótipos, que são calculados diretamente, o desempenho de um neurônio é computado como a média do *fitness* das 5 melhores redes (protótipos) nas quais ele participa.

Este esquema de *fitness* compartilhado dá uma estimativa de quanto o neurônio coopera com outros neurônios da população para a solução do problema proposto. A relação entre neurônios e protótipos das redes neurais é cooperativa. A recompensa dos indivíduos, baseada no quanto eles cooperam entre si, resulta na garantia de diversidade e na possibilidade de decomposição do problema em módulos mais simples.

Para Potter & De Jong (2000), a evolução de múltiplas populações geneticamente isoladas representa uma solução simples para problemas com restrições. Além disso, o isolamento genético das populações, garantido pela ausência de troca de material genético entre indivíduos de espécie diferentes, tende a preservar a diversidade populacional, pois guia as espécies para a convergência dentro de nichos distintos, ao invés de subsistirem dentro de um mesmo nicho. A recompensa para os indivíduos, baseada no quanto eles cooperam com representantes de outras espécies, imprime uma pressão para que façam contribuições únicas para a solução do problema. No entanto, é importante salientar que a existência de separação genética entre as espécies não impede a interação de seus indivíduos. Indivíduos de espécies distintas emergem de processos reprodutivos independentes, mas atuam de forma cooperativa em um modelo de domínio comum para formar a solução do problema.

No trabalho apresentado por Potter & De Jong (2000), um modelo de co-evolução cooperativa é apresentado como forma de se obterem algoritmos evolutivos complexos, nos quais um número apropriado de sub-componentes interdependentes (cobrindo nichos diferentes) evolui atingindo um nível de generalidade desejado. A estrutura geral deste modelo co-evolutivo está ilustrada na figura 2.12. Embora o esquema mostre três espécies diferentes, num processo de co-evolução genérico, o número real de espécies no ecossistema pode ser maior ou menor. Cada espécie evolui

sua própria população e é capaz de se adaptar ao ambiente através de sucessivas aplicações de um algoritmo evolutivo (AE). A figura 2.12 evidencia a etapa de avaliação de desempenho (cálculo do *fitness*) do AE na perspectiva de cada uma das três espécies mostradas. No processo de avaliação dos indivíduos de uma espécie, as cooperações se estabelecem através de representantes de cada uma das espécies restantes. A escolha dos representantes pode se dar de diferentes formas: escolha determinística, na qual os representantes são sempre os melhores indivíduos de cada espécie; escolha não-determinística, proporcional ao *fitness*; escolha baseada em critérios de distância, na qual os indivíduos que compartilham uma vizinhança pré-determinada são eleitos como representantes no processo cooperativo.

A abordagem proposta nesta tese é baseada em um modelo co-evolutivo com componentes de hierarquia e cooperação e tem como maiores inspirações os paradigmas propostos por Moriarty & Miikkulainen (1998) e Potter & De Jong (2000). Assim como no modelo proposto por Potter & De Jong (2000), na abordagem co-evolutiva que está sendo proposta, as cooperações também se estabelecem através de representantes de cada uma das espécies restantes, só que esta cooperação respeita uma hierarquia pré-definida (indivíduos de uma população em um nível hierarquicamente inferior entram na composição de indivíduos em níveis hierarquicamente superiores).

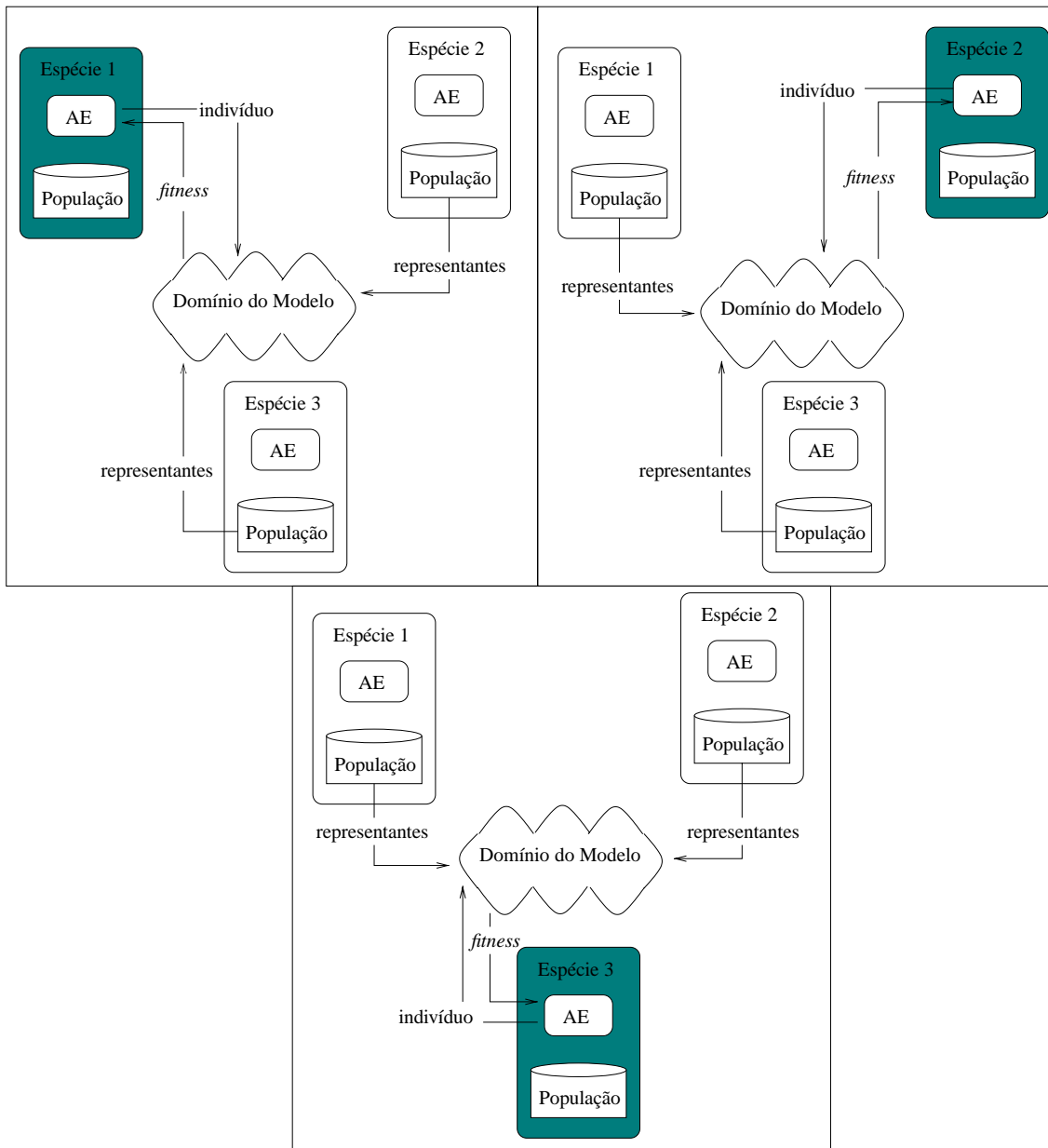


Figura 2.12: Exemplo da co-evolução cooperativa proposta por Potter & De Jong (2000).

Capítulo 3

Sistemas Nebulosos

O conceito de conjunto nebuloso (*fuzzy set*) foi introduzido por Zadeh (1965) e, desde então, os sistemas baseados em conjuntos nebulosos, ou sistemas nebulosos, vêm avançando tanto do ponto de vista teórico quanto prático.

Os termos conjuntos nebulosos e lógica nebulosa têm sido freqüentemente utilizados como sinônimos. Entretanto, é importante salientar que estas duas nomenclaturas tratam aspectos distintos da teoria.

- **Conjuntos Nebulosos:** A designação de conjuntos nebulosos é geralmente usada para descrever toda a teoria de conjuntos nebulosos que engloba, inclusive, alguns conceitos de lógica nebulosa. A teoria de conjuntos nebulosos, vista como uma extensão da teoria de conjuntos clássicos, está associada aos conceitos básicos de funções de pertinência, operações com conjuntos nebulosos, números nebulosos, relações nebulosas, regras nebulosas, regra composicional de inferência, etc.
- **Lógica Nebulosa:** A lógica nebulosa pode ser vista como uma extensão das lógicas bi-valores e multi-valores (Resher, 1969; Lukasiewicz, 1970). Ao invés de considerar apenas dois valores-verdade (Verdadeiro=0 e Falso=1) como é o caso da lógica bi-valores, ou um intervalo unitário como no caso da lógica multi-valores, a lógica nebulosa pressupõe que os valores-verdade são conjuntos nebulosos definidos no intervalo $[0, 1]$. A definição de regras nebulosas com base em implicações lógicas permite o uso do termo lógica nebulosa para descrever a computação baseada em regras nebulosas. A teoria associada à lógica nebulosa requer a definição de conceitos distintos, como valor-verdade, proposições e implicações nebulosas.

A utilização de conjuntos nebulosos e de um tipo de computação baseada na regra composicional de inferência, com seus operadores associados, resulta em sistemas de decisão, estruturados no formato de uma base de regras nebulosas, adequados para a implementação de processos dedutivos. Estes sistemas, conhecidos como sistemas de inferência nebulosa ou simplesmente sistemas nebulosos, dependem da especificação de uma série de elementos, que incluem a quantidade e o tipo de regras nebulosas, os parâmetros das funções de pertinência, a semântica das regras que participam do raciocínio aproximado e os operadores do mecanismo de inferência utilizado para obter uma saída, a partir dos dados de entrada. A utilização de sistemas nebulosos na forma de raciocínio aproximado representa um marco na tecnologia da informação, pois os sistemas baseados em regras nebulosas possuem grande habilidade para expressar a ambigüidade e subjetividade presentes no raciocínio humano (Pedrycz & Gomide, 1998).

Segundo discussão apresentada por Lin & Lee (1996), o uso de sistemas nebulosos é indicado quando:

- o modelo matemático do sistema não existe, ou existe, mas é de difícil codificação;
- o modelo matemático é muito complexo para ser rapidamente avaliado em tempo real ou requer muita memória para ser implementado fisicamente;
- as variáveis do fenômeno em estudo são contínuas;
- existe alto nível de ruído no ambiente e/ou apenas sensores de baixa precisão disponíveis;
- o processo envolve a interação com um operador humano, ou especialista, preparado para especificar os parâmetros do conjunto de regras a ser utilizado no sistema nebuloso.

Acrescentem-se a esses itens a possibilidade de um projeto automático para o ajuste de parâmetros na ausência do especialista e o interesse pela representação do conhecimento por uma base de regras lingüísticas, e está criado o cenário ideal para a utilização dos sistemas nebulosos. Estudos recentes destes sistemas apontam para aplicações em áreas diversificadas que vão desde controle, reconhecimento de padrão e sistemas de base de conhecimento, passando por visão computacional e chegando em vida artificial. Um número significativo de implementações práticas vem consolidando os sistemas nebulosos não só em instalações industriais, mas também em muitos produtos manufaturados de uso diário (Pedrycz & Gomide, 1998).

Muitas das metodologias para análise de base de dados pressupõem dados precisos e medidas exatas. Mas no cenário prático isto acontece em raríssimas situações. A maioria das medidas

armazenadas deve ser interpretada, na realidade, como intervalos cujos comprimentos dependem da precisão dos dispositivos de medição (Berthold & Hand, 1999). A teoria dos sistemas nebulosos permite que se trabalhe com objetos de bordas ou fronteiras difusas e, por isso, os sistemas nebulosos são também conhecidos como sistemas difusos. Esta propriedade, fundamentada pelos conceitos básicos dos conjuntos nebulosos, imprime a principal característica dos sistemas nebulosos: a capacidade de lidar com informações imprecisas.

O conceito de imprecisão associado às informações processadas pelos sistemas nebulosos é usualmente confundido com a incerteza relacionada à teoria da probabilidade. A principal diferença entre a imprecisão nos sistemas nebulosos e a incerteza probabilística é que a primeira trata de possibilidade (plausibilidade) determinística, ao passo que a segunda lida com a verossimilhança de eventos não-determinísticos ou estocásticos (Berthold & Hand, 1999). A incerteza que está por trás do conceito de probabilidade geralmente se relaciona com a ocorrência de um fenômeno simbolizado pelo conceito de aleatoriedade, como, por exemplo, quando se tenta determinar “a probabilidade de alguma pessoa escolhida aleatoriamente dentro de uma população ser jovem”, ou “a probabilidade do próximo carro que cruza a avenida ter velocidade alta”. Por outro lado, a imprecisão nos conjuntos nebulosos está relacionada à ambigüidade encontrada na definição de conceitos lingüísticos como “pessoa jovem” ou “velocidade alta”.

Este capítulo apresenta os conceitos fundamentais para o entendimento e utilização dos sistemas nebulosos. Na seção 3.1 são apresentados conceitos básicos dos conjuntos nebulosos, como: funções de pertinência, operações com conjuntos nebulosos e relações nebulosas. A seção 3.2 traz um tipo de computação que se baseia em conceitos distintos, como: variável lingüística, regras nebulosas e regra composicional de inferência, permitindo-se assim, que se crie o elo de ligação entre os conjuntos nebulosos e o raciocínio aproximado. A seção 3.3 descreve os sistemas baseados em conjuntos nebulosos, ou sistemas nebulosos, apresentando os diferentes mecanismos de inferência utilizados por estes sistemas para produzir saídas a partir dos dados de entrada. Esta seção descreve ainda os dois modelos de sistemas nebulosos mais comumente utilizados em aplicações práticas: o modelo de Mamdani e o modelo Takagi-Sugeno.

3.1 Introdução aos Conjuntos Nebulosos

A noção de conjunto ocorre freqüentemente quando se tenta organizar, resumir e generalizar o conhecimento a respeito de objetos. Seja X uma coleção de objetos denominados genericamente

por x . Então, um conjunto A é definido por uma coleção de pares ordenados

$$A = \{(x, \mu_A(x)) \mid x \in \mathbf{X}\} .$$

A função $\mu_A(x)$ é denominada função de pertinência e determina com que grau um objeto x pertence a um conjunto A e \mathbf{X} é chamado de universo. Em conjuntos clássicos, apenas dois valores para $\mu_A(x)$ são permitidos: o elemento pertence ou não pertence a um determinado conjunto. Na teoria dos conjuntos nebulosos a transição entre pertencer e não pertencer é gradual.

Como exemplo, considere $\mathbf{X} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ uma coleção de números inteiros. Seja A um conjunto nebuloso que define “números inteiros próximos a 5”, dado por

$$A = \{(1, 0), (2, 0.4), (3, 0.6), (4, 0.8), (5, 1), (6, 0.8), (7, 0.6), (8, 0.4), (9, 0)\} .$$

Então, o conjunto nebuloso A pode ser definido como uma coleção de objetos com valores de pertinência variando entre 0 (exclusão completa) e 1 (pertinência completa). Os conjuntos nebulosos representam, portanto, uma generalização dos conjuntos clássicos. O conceito de função de pertinência, fundamental para a teoria dos conjuntos nebulosos, será discutido na seção a seguir.

3.1.1 Funções de Pertinência

Toda a teoria dos conjuntos está baseada no conceito de pertinência. Um conjunto nebuloso é definido pela função de pertinência $\mu_A(x)$ que estabelece para cada x um grau de pertinência ao conjunto A , com $\mu_A(x) \in [0, 1]$. Os conjuntos clássicos podem ser vistos como um caso particular dos conjuntos nebulosos, no qual apenas os limites do intervalo são utilizados na definição da função de pertinência: $\mu_A(x) \in \{0, 1\}$, $x \in \mathbf{X}$.

Por exemplo, considerando-se o universo contínuo $\mathbf{X} \in \mathfrak{R}$, o conceito de “números próximos a 5” pode ser expresso de forma diferente dependendo da definição da função de pertinência associada”

- “números próximos a 5” na concepção clássica

$$\mu_A(x) = \begin{cases} 0 & \text{se } x \leq 4.5 \\ 1 & \text{se } 4.5 < x \leq 5.5 \\ 0 & \text{se } x > 5.5 ; \end{cases}$$

- “números próximos a 5” na concepção de conjuntos nebulosos

$$\mu_A(x) = \begin{cases} 0 & \text{se } x \leq 4.5 \\ \frac{x-4.5}{0.5} & \text{se } 4.5 < x \leq 5 \\ \frac{5.5-x}{0.5} & \text{se } 5 < x \leq 5.5 \\ 1 & \text{se } x > 5.5 . \end{cases}$$

A figura 3.1 ilustra as diferenças entre as funções de pertinência no caso de conjuntos clássicos e nebulosos.

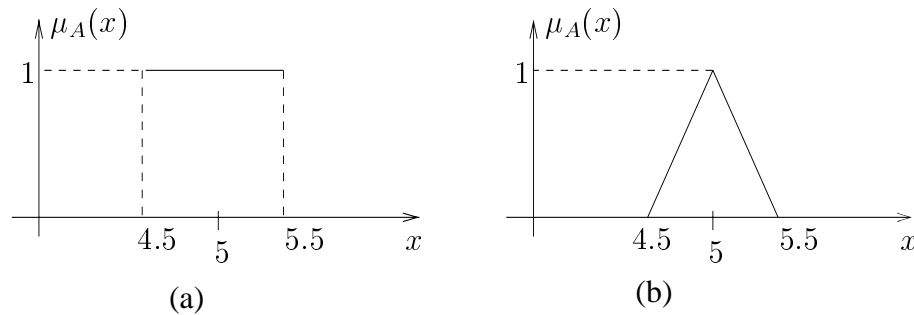


Figura 3.1: Funções de pertinência: (a) Conjuntos clássicos, (b) Conjuntos nebulosos.

Em geral, o formato das funções de pertinência é restrito a uma certa classe de funções, representadas por alguns parâmetros específicos. Os formatos mais comuns são: triangular, trapezoidal e Gaussiana. Além dos formatos tradicionais existe uma forma bastante utilizada em aplicações práticas: o conjunto unitário (*singleton*).

- Função Triangular: parâmetros (a, m, b) , com $a \leq m \leq b$

$$\mu_A(x) = \begin{cases} 0 & \text{se } x \leq a \\ \frac{x-a}{m-a} & \text{se } a < x \leq m \\ \frac{b-x}{b-m} & \text{se } m < x \leq b \\ 0 & \text{se } x > b ; \end{cases}$$

- Função Trapezoidal: parâmetros (a, m, n, b) , com $a \leq m, n \leq b$ e $m < n$

$$\mu_A(x) = \begin{cases} 0 & \text{se } x \leq a \\ \frac{x-a}{m-a} & \text{se } a < x \leq m \\ 1 & \text{se } m < x \leq n \\ \frac{b-x}{b-n} & \text{se } n < x \leq b \\ 0 & \text{se } x > b ; \end{cases}$$

- Função Gaussiana: parâmetros (m, σ_k) , com $\sigma_k > 0$

$$\mu_A(x) = \exp^{-\sigma_k(x-m)^2} ;$$

- Conjunto Unitário (*singleton*): parâmetros (m, h)

$$\mu_A(x) = \begin{cases} h & \text{se } x = m \\ 0 & \text{caso contrário .} \end{cases}$$

A figura 3.2 ilustra os formatos descritos acima, especificando os parâmetros associados a cada caso.

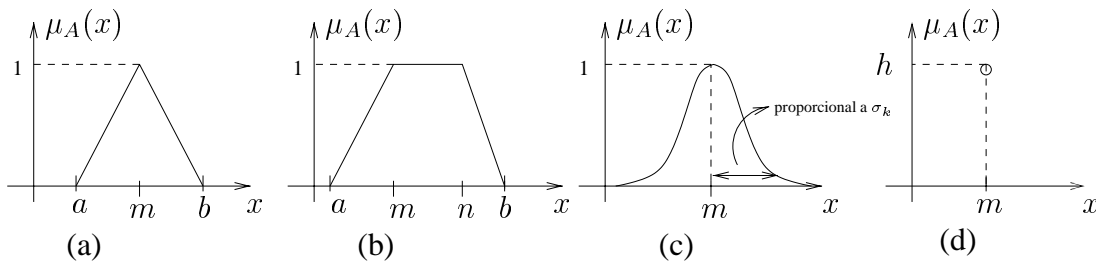


Figura 3.2: Diferentes formatos das funções de pertinência: (a) Triangular, (b) Trapezoidal, (c) Gaussiana, (d) *Singleton*.

A escolha do formato mais adequado nem sempre é óbvia, podendo inclusive não estar ao alcance do conhecimento de um especialista para a aplicação em questão. No entanto, existem sistemas nebulosos cujos parâmetros das funções de pertinência são completamente definidos pelo especialista. Nestes casos, a escolha de funções triangulares e trapezoidais é mais comum porque a idéia de se definirem regiões de pertinência total, média e nula é mais intuitiva do que a especificação do valor modal (m) e dispersão (σ_k), conceitos ligados às funções Gaussianas. Entretanto, existe cada

vez mais uma tendência de sistemas nebulosos adaptativos nos quais os parâmetros das funções de pertinência são ajustados, no sentido de otimizar algum objetivo definido, a partir dos dados de entrada-saída. Em sistemas que ajustam estes parâmetros pelo método do gradiente, por exemplo, a função Gaussiana é bastante utilizada, devido às suas propriedades matemáticas de continuidade e diferenciabilidade (Jang *et al.*, 1997).

Neste trabalho, um outro tipo de função de pertinência associada a uma variável qualquer x será utilizado para indicar a irrelevância desta variável na regra nebulosa. Esta condição irrelevante (*don't care condition* (Ishibuchi & Nakashima, 1999)) será representada (em proposições que utilizam o E como agregação) por uma função denominada função de irrelevância, definida como:

$$\text{Função de Irrelevância : } \mu_A(x) = 1 \quad \forall x \in \mathbf{X} .$$

3.1.2 Definições Básicas em Conjuntos Nebulosos

Embora a função de pertinência, especificada pelos parâmetros que definem o seu formato, seja usualmente utilizada para representar um conjunto nebuloso, existem outros parâmetros que podem ser usados para caracterizar os conjuntos nebulosos:

- suporte (S_A): conjunto dos elementos do universo, para os quais o grau de pertinência é maior do que zero.

$$S_A = \{x \mid \mu_A(x) > 0\} ;$$

- núcleo (*core*)(N_A): conjunto dos elementos do universo com grau de pertinência igual a 1.

$$N_A = \{x \mid \mu_A(x) = 1\} ;$$

- altura (H_A): valor máximo da função de pertinência, dado por

$$H_A = \sup_x \{\mu_A(x)\} .$$

Uma outra forma de representação é via α -cortes. Esta forma permite representar um conjunto nebuloso como uma família de outros conjuntos nebulosos particulares e determina a base do teorema da representação (Pedrycz & Gomide, 1998).

- α -corte : é o conjunto dos elementos do universo para os quais os graus de pertinência são superiores ou iguais a α .

$$C_{\alpha_A} = \{x \mid \mu_A(x) \geq \alpha\} .$$

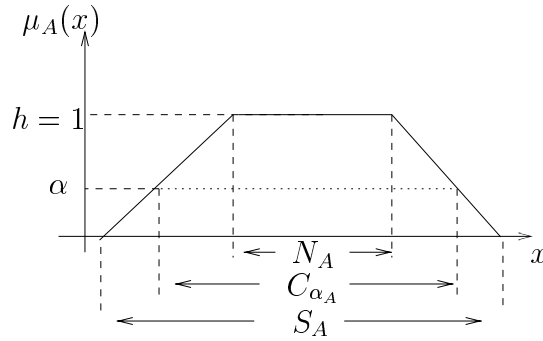


Figura 3.3: Suporte (S_A), α -corte (C_{α_A}), núcleo (N_A), altura ($H_A = h$).

A figura 3.3 mostra estas características.

Outras definições importantes são:

- **Conjunto Normal:** um conjunto nebuloso é dito normal se sua altura é igual a 1, ou seja, $H_A = 1$. Caso contrário é chamado de subnormal. O núcleo (*core*) de um conjunto subnormal é um conjunto vazio.
- **Conjunto Convexo:** um conjunto nebuloso é convexo se sua função de pertinência é tal que:

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

para todo $x_1, x_2 \in \mathbf{X}$ e $\lambda \in [0, 1]$.

- **Subconjunto Nebuloso:** se $\mu_A(x) \leq \mu_B(x)$, para todo x então $A \subseteq B$, ou seja, A é subconjunto de B . Uma propriedade derivada da equação anterior determina que

$$\text{se } A \subseteq B \text{ e } B \subseteq C \Rightarrow A \subseteq C .$$

A figura 3.4 mostra exemplos de conjuntos e subconjuntos nebulosos normais, subnormais, convexos e não-convexos.

3.1.3 Operações com Conjuntos Nebulosos

Existem várias operações que podem ser aplicadas a conjuntos nebulosos. Há operações com argumento único, que modificam o formato da função de pertinência. Dentre as mais comuns podemos citar a normalização (que converte um conjunto subnormal em um normal), a concentração (que diminui os valores da função de pertinência), a dilatação (que aumenta os valores da função de

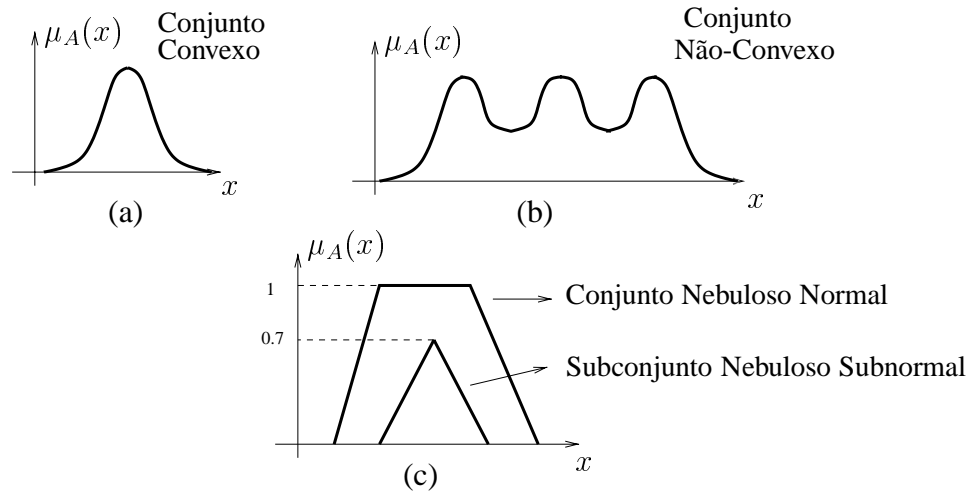


Figura 3.4: Convexidade e normalidade em conjuntos nebulosos.

pertinência) e a intensificação de contraste (que dilata o conjunto para valores de pertinência acima de 0.5 e concentra o conjunto para valores abaixo de 0.5) (Pedrycz & Gomide, 1998). Há operações de múltiplos argumentos, que envolvem a combinação, agregação e comparação de dois ou mais conjuntos nebulosos. A maioria destas operações é derivada da teoria de conjuntos clássicos. A seguir, serão descritas a operação de complemento e as operações de múltiplos argumentos mais comumente aplicadas a conjuntos nebulosos.

União, Interseção e Complemento

A união, interseção e complemento são as operações essenciais realizadas em conjuntos clássicos. A tabela 3.1 resume algumas propriedades básicas destas operações, considerando os conjuntos clássicos A e B definidos num universo X .

Com base na teoria dos conjuntos clássicos, Zadeh (1965) definiu as operações de união, interseção e complemento para conjuntos nebulosos, a partir da função de pertinência:

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

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

$$\text{Complemento:} \quad \mu_{(\bar{A})}(x) = 1 - \mu_A(x) .$$

Devido ao isomorfismo entre a teoria dos conjuntos e a lógica proposicional bi-valores, a interseção e a união podem ser identificadas pela conjunção (E) e pela disjunção (OU), respectivamente. As-

Tabela 3.1: Propriedades das operações com conjuntos clássicos.

Comutatividade	$A \cup B = B \cup A, A \cap B = B \cap A$
Associatividade	$(A \cup B) \cup C = A \cup (B \cup C)$ $(A \cap B) \cap C = A \cap (B \cap C)$
Distributividade	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
Idempotência	$A \cup A = A, A \cap A = A$
Condições Limites	$A \cup \emptyset = A, A \cup \mathbf{X} = \mathbf{X}$ $A \cap \emptyset = \emptyset, A \cap \mathbf{X} = A$
Involução	$\overline{\overline{A}} = A$
Lei da Contradição	$A \cap \overline{A} = \emptyset$
Lei da Exclusão do Meio	$A \cup \overline{A} = \mathbf{X}$
Lei de Morgan	$\overline{A \cup B} = \overline{A} \cap \overline{B}$ $\overline{A \cap B} = \overline{A} \cup \overline{B}$

sim, as operações de conjunção e disjunção podem ser representadas pelos operadores \wedge e \vee (Pedrycz & Gomide, 1998). A figura 3.5 mostra as operações de união, interseção e complemento aplicadas a dois conjuntos nebulosos A e B .

Normas e Co-normas Triangulares

As normas e co-normas triangulares formam uma classe geral de operadores de união e interseção, com características de comutatividade, associatividade e monotonicidade, atendendo ainda as condições de contorno, conforme mostrado na tabela 3.1 (Pedrycz & Gomide, 1998). Diferente da união e interseção, que trabalham com conjuntos definidos num mesmo universo, as operações baseadas em normas e co-normas triangulares podem operar conjuntos em universos distintos.

Sejam A e B dois conjuntos nebulosos definidos nos universos \mathbf{X} e \mathbf{Y} , respectivamente, e a e b valores de pertinência dados por $a = \mu_A(x)$ e $b = \mu_B(y)$. Então, as normas e co-normas triangulares (norma-t e norma-s) podem ser definidas como:

- Norma-t

operador de dois argumentos $\mathbf{t} : [0, 1]^2 \rightarrow [0, 1]$ que satisfaz as seguintes condições:

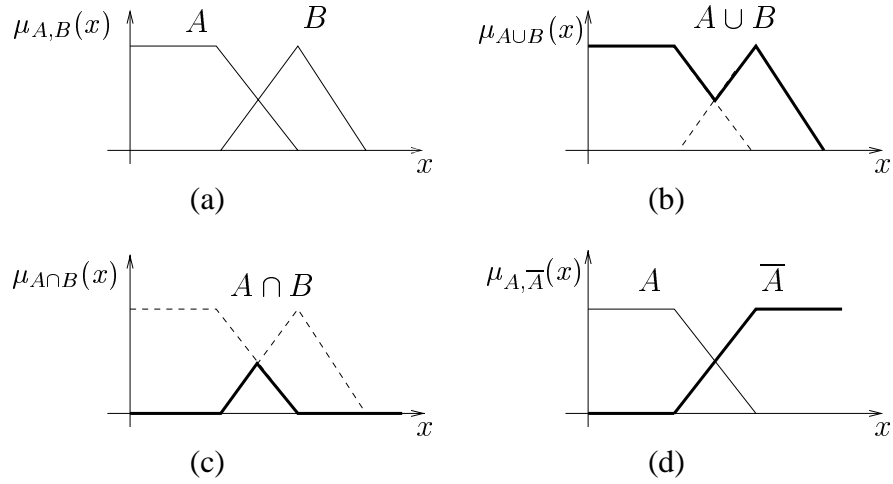


Figura 3.5: Operações com conjuntos nebulosos: (a) Conjuntos A e B , (b) União, (c) Interseção (d) Complemento.

- * Comutatividade: $a \mathbf{t} b = b \mathbf{t} a$;
- * Associatividade: $a \mathbf{t} (b \mathbf{t} c) = (a \mathbf{t} b) \mathbf{t} c$;
- * Monotonicidade: se $a \leq b$ e $c \leq d$, então, $a \mathbf{t} c \leq b \mathbf{t} d$;
- * Condições de Contorno: $0 \mathbf{t} a = 0$, $1 \mathbf{t} a = a$.

Exemplos de normas-t:

$$a \mathbf{t}_1 b = \frac{1}{1 + \sqrt[p_t]{\frac{1-a}{a} + \frac{1-b}{b}}}, \quad p_t > 0$$

$$a \mathbf{t}_2 b = \max[0, (1 + p_t)(a + b - 1) - p_t ab], \quad p_t \geq -1$$

$$a \mathbf{t}_3 b = 1 - \min(1, \sqrt[p_t]{(1-a)^{p_t} + (1-b)^{p_t}}), \quad p_t > 0$$

$$a \mathbf{t}_4 b = ab$$

$$a \mathbf{t}_5 b = \frac{ab}{p_t + (1-p_t)(a+b-ab)}, \quad p_t \geq 0$$

$$a \mathbf{t}_6 b = \frac{1}{\sqrt[p_t]{\frac{1}{a^{p_t}} + \frac{1}{b^{p_t}} - 1}}$$

$$a \mathbf{t}_7 b = \sqrt[p_t]{\max[0, a^{p_t} + b^{p_t} - 1]}, \quad p_t > 0$$

$$a \mathbf{t}_8 b = \begin{cases} a & \text{se } b = 1, \\ b, & \text{se } a = 1, \\ 0, & \text{caso contrário.} \end{cases}$$

$$a \mathbf{t}_9 b = \min[a, b]$$

- Norma-s

operador de dois argumentos $s : [0, 1]^2 \rightarrow [0, 1]$ que satisfaz as seguintes condições:

- * Comutatividade: $a s b = b s a$;
- * Associatividade: $a s (b s c) = (a s b) s c$;
- * Monotonicidade: se $a \leq b$ e $c \leq d$, então, $a s c \leq b s d$;
- * Condições de contorno: $0 s a = a$, $1 s a = 1$;

Exemplos de normas-s:

$$a s_1 b = \frac{1}{1 + \sqrt[p_t]{(a/(1-a))^{p_t} + (b/(1-b))^{p_t}}}, \quad p_t > 0$$

$$a s_2 b = \min[1, a + b - p_t ab], \quad p_t \geq 0$$

$$a s_3 b = \min(1, \sqrt[p_t]{a^{p_t} + b^{p_t}}), \quad p_t > 0$$

$$a s_4 b = a + b - ab$$

$$a s_5 b = \frac{a+b-ab-(1-p_t)ab}{1-(1-p_t)ab}, \quad p_t \geq 0$$

$$a s_6 b = 1 - \frac{1}{\sqrt[p_t]{\frac{1}{(1-a)^{p_t}} + \frac{1}{(1-b)^{p_t}} - 1}}$$

$$a s_7 b = 1 - \max \left[0, \sqrt[p_t]{(1-a)^{p_t} + (1-b)^{p_t}} - 1 \right], \quad p_t > 0$$

$$a s_8 b = \begin{cases} a & \text{se } b = 0, \\ b, & \text{se } a = 0, \\ 1, & \text{caso contrário.} \end{cases}$$

$$a s_9 b = \max[a, b]$$

Em geral, as normas t e s não satisfazem, necessariamente, as demais propriedades válidas para os conjuntos clássicos (tabela 3.1). Por exemplo, a lei da contradição e o princípio da exclusão do meio são características presentes apenas nas normas t_2 e s_2 , considerando-se $p_t = 0$. Para todas as outras normas listadas anteriormente, essas duas condições não se aplicam. Para exemplificar, considere a norma- $t = t_9 = \min$ e a co-norma- $s = s_9 = \max$, e um valor de pertinência $a = \mu_A(x) = 0.5$. Então:

$$A \cap \bar{A} = a t_9 (1-a) = \min[a, 1-a] = 0.5 \neq \emptyset;$$

$$A \cup \bar{A} = a s_9 (1-a) = \max[a, 1-a] = 0.5 \neq \mathbf{X}.$$

O princípio da idempotência só se aplica às as normas *min* e *max*.

$$a \mathbf{t}_9 a = \min[a, a] = a ;$$

$$a \mathbf{s}_9 a = \max[a, a] = a .$$

Para as outras normas, apenas podemos garantir que:

$$\underbrace{a \mathbf{t} a \mathbf{t} a \mathbf{t} a \cdots a \mathbf{t} a}_{n \text{ vezes}} \leq \underbrace{a \mathbf{t} a \mathbf{t} a \cdots a \mathbf{t} a}_{n - 1 \text{ vezes}} ;$$

$$\underbrace{a \mathbf{s} a \mathbf{s} a \mathbf{s} a \cdots a \mathbf{s} a}_{n \text{ vezes}} \geq \underbrace{a \mathbf{s} a \mathbf{s} a \cdots a \mathbf{s} a}_{n - 1 \text{ vezes}} .$$

Assim como a idempotência, a propriedade de distributividade está presente apenas nas normas **min** e **max** (ver Klement *et al.* (2000) para maiores detalhes em classificação de normas e co-normas triangulares).

As normas **t** e **s** são consideradas operadores de agregação, pois podem ser usadas para combinar uma coleção de conjuntos nebulosos para produzir um único conjunto, também nebuloso. A figura 3.6 mostra o resultado da agregação (conjunção e disjunção) de dois conjuntos nebulosos *A* e *B* em *X* e *Y*, respectivamente, dada por duas normas duais (**t**₂ e **s**₂).

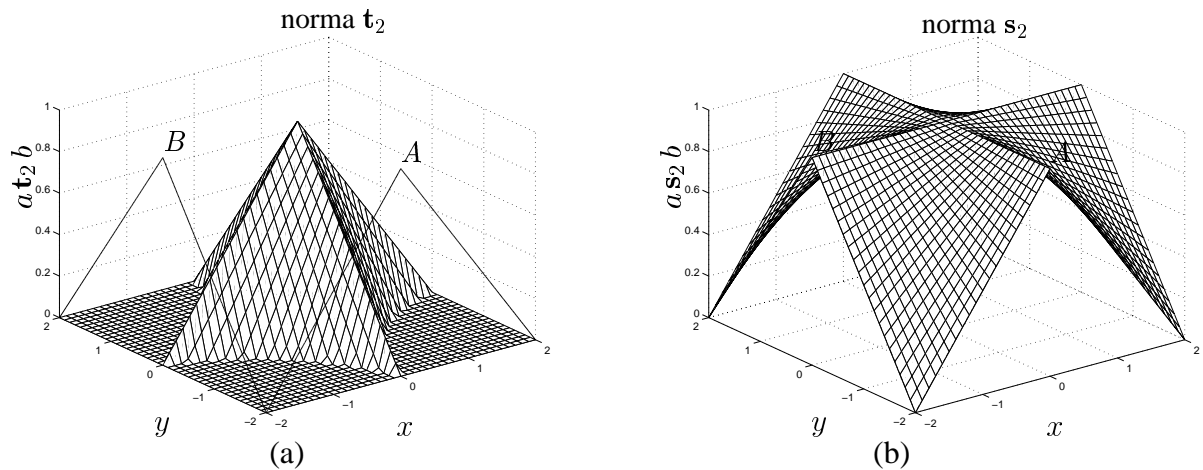


Figura 3.6: Agregação por normas **t** e **s** : (a) conjunção dada pela norma-**t**: $a \mathbf{t}_2 b$ (b) disjunção dada pela norma-**s**: $a \mathbf{s}_2 b$, onde $a = \mu_A(x)$, $b = \mu_B(y)$ e $p_t = 2$.

Outras Operações

Embora as normas e co-normas triangulares possam realizar a operação de agregação entre conjuntos nebulosos, existem operadores de agregação específicos, como por exemplo os operadores

compensatórios propostos por Zimmermann & Zysno (1980), nos quais a agregação é feita com base em uma combinação de conjunção (E) e disjunção (OU). Outros exemplos de operadores de agregação propostos na literatura são os operadores de soma simétrica, como o apresentado por Dubois & Prade (1980); os operadores de média generalizada, como o discutido por Dyckhoff & Pedrycz (1984); ou ainda o operador de média ponderada ordenada (OWA), proposto por Yager (1988).

Além da agregação, as operações envolvem ainda comparações que resultam em medidas de distância, igualdade, possibilidade, necessidade e compatibilidade entre conjuntos nebulosos (para maiores detalhes, ver Dubois & Prade (1980); Klement *et al.* (2000); Pedrycz & Gomide (1998)). Dentre essas medidas, a possibilidade e a necessidade resultam em interpretações bastante interessantes. A medida de possibilidade quantifica o grau de sobreposição entre dois conjuntos. A medida de necessidade quantifica o grau de inclusão de um conjunto em outro. A seguir, são definidas estas duas medidas aplicadas a dois conjuntos nebulosos A e B no mesmo universo \mathbf{X} :

- Possibilidade: $\text{Poss}(A, B) = \sup_{x \in \mathbf{X}} [\min(\mu_A(x), \mu_B(x))]$;
- Necessidade: $\text{Nec}(A, B) = \inf_{x \in \mathbf{X}} [\max(\mu_A(x), 1 - \mu_B(x))]$.

3.1.4 Relações Nebulosas

As relações nebulosas são generalizações das relações tradicionais. Por serem mais gerais do que as funções, as relações (tradicionais ou nebulosas) permitem que as dependências entre as variáveis envolvidas sejam capturadas sem que nenhuma caracterização direcional particular seja fixada, ou seja, não há domínio e contra-domínio (Pedrycz & Gomide, 1998).

Sejam \mathbf{X} e \mathbf{Y} dois universos quaisquer. Uma relação nebulosa R é um conjunto nebuloso definido em $\mathbf{X} \times \mathbf{Y}$ que associa a cada elemento de $\mathbf{X} \times \mathbf{Y}$ um grau de pertinência definido no intervalo unitário, ou seja, $R : \mathbf{X} \times \mathbf{Y} \rightarrow [0, 1]$.

Por exemplo, uma relação nebulosa, vista como uma generalização do produto cartesiano clássico $\mathbf{X} \times \mathbf{Y} \rightarrow \{0, 1\}$, é dada por:

$$R = \{((x, y), \mu_R(x, y)) \mid (x, y) \in \mathbf{X} \times \mathbf{Y}\} .$$

Sejam, $\mathbf{X} = \mathbf{Y} = \mathfrak{R}^+$ e $R = \text{“}y \text{ muito maior do que } x\text{”}$. A função de pertinência da relação nebulosa R pode ser definida (subjetivamente) como mostra a figura 3.7.

Quando as relações nebulosas envolvem universos discretos, elas podem ser representadas por matrizes ou grafos. Os nós deste grafo são os elementos de \mathbf{X} e \mathbf{Y} , enquanto que as ligações

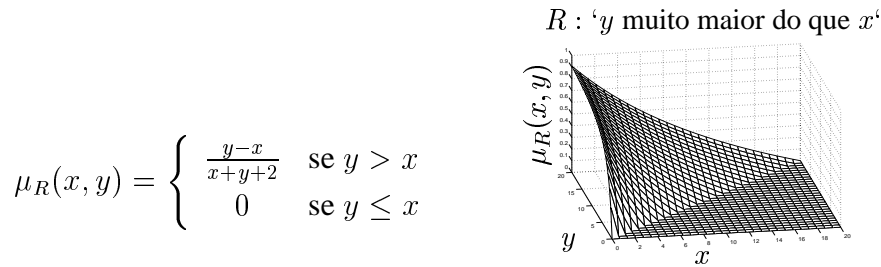


Figura 3.7: Exemplo de relação nebulosa num universo contínuo.

correspondem aos graus de pertinência. Por exemplo, a figura 3.8 mostra um exemplo de uma relação nebulosa discreta representada por uma matriz e um grafo.

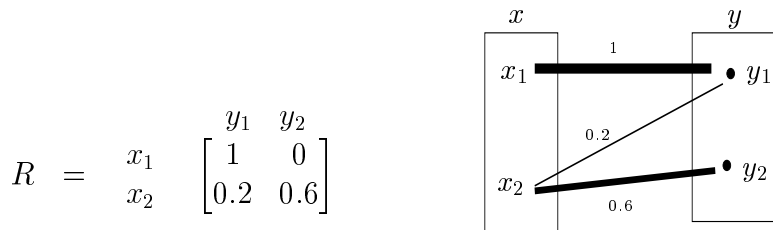


Figura 3.8: Exemplo de relação nebulosa num universo discreto.

As relações nebulosas mencionadas acima são exemplos de relações nebulosas definidas em espaços bidimensionais. Conforme discutido em Jang *et al.* (1997), existem outros exemplos de relações nebulosas envolvendo duas variáveis ou objetos:

- x está próximo a y (x e y são números);
- x e y são parecidos (x e y são pessoas ou objetos);
- se x é grande, então y é pequeno (x é uma variável observável e y é uma ação correspondente).

A última relação, representada pela expressão “se x é A então y é B ”, é conhecida como regra nebulosa, sendo comumente utilizada em bases de regras de sistemas nebulosos. Esta relação será detalhada na seção 3.2.2.

Relações nebulosas multidimensionais, também chamadas de relações nebulosas n -árias, podem ser obtidas por generalização das relações nebulosas bidimensionais:

$$R : \mathbf{X}_1 \times \mathbf{X}_2 \cdots \times \mathbf{X}_n \rightarrow [0, 1] .$$

Operações com Relações Nebulosas

As relações nebulosas definidas em produtos cartesianos idênticos podem ser combinadas através de operações de união, interseção e complemento. Sejam R_1 e R_2 duas relações nebulosas em $\mathbf{X} \times \mathbf{Y}$:

$$R_1 = \{((x, y), \mu_{R_1}(x, y)) \mid (x, y) \in \mathbf{X} \times \mathbf{Y}\} ;$$

$$e$$

$$R_2 = \{((x, y), \mu_{R_2}(x, y)) \mid (x, y) \in \mathbf{X} \times \mathbf{Y}\} .$$

Então, as operações de união, interseção e complemento para R_1 e R_2 são dadas por:

$$\text{União:} \quad \mu_{(R_1 \cup R_2)}(x, y) = \mu_{R_1}(x, y) \mathbf{s} \mu_{R_2}(x, y) ;$$

$$\text{Interseção:} \quad \mu_{(R_1 \cap R_2)}(x, y) = \mu_{R_1}(x, y) \mathbf{t} \mu_{R_2}(x, y) ;$$

$$\text{Complemento:} \quad \mu_{(\overline{R_1})}(x, y) = 1 - \mu_{R_1}(x, y) .$$

Composição de Relações Nebulosas

As relações nebulosas definidas em espaços distintos podem ser combinadas utilizando-se diferentes operadores de composição. As mais comuns são:

- composição sup-t: $R = G \circ W$

$$\mu_R(x, y) = \sup_{z \in \mathbf{Z}} [\mu_G(x, z) \mathbf{t} \mu_W(z, y)] .$$

As composições sup-t mais conhecidas são as que utilizam o *min* e o produto algébrico como normas-t, sendo denominadas de composição *max-min* e *max-prod*, respectivamente.

A seguir, será ilustrado um exemplo de composição sup-t entre duas relações nebulosas discretas G e W , onde

$$G = \begin{array}{c} x_1 \\ x_2 \end{array} \begin{array}{ccc} z_1 & z_2 & z_3 \\ \left[\begin{array}{ccc} 0.2 & 0.6 & 0 \\ 0 & 0.9 & 0.7 \end{array} \right] \end{array} \quad W = \begin{array}{c} z_1 \\ z_2 \\ z_3 \end{array} \begin{array}{cc} y_1 & y_2 \\ \left[\begin{array}{cc} 0.3 & 0.9 \\ 0.1 & 0.5 \\ 0 & 0.9 \end{array} \right] \end{array} .$$

A representação por um grafo da composição $R = G \circ W$ explicitando-se a composição das variáveis (x_2, y_1) , é mostrada na figura 3.9.

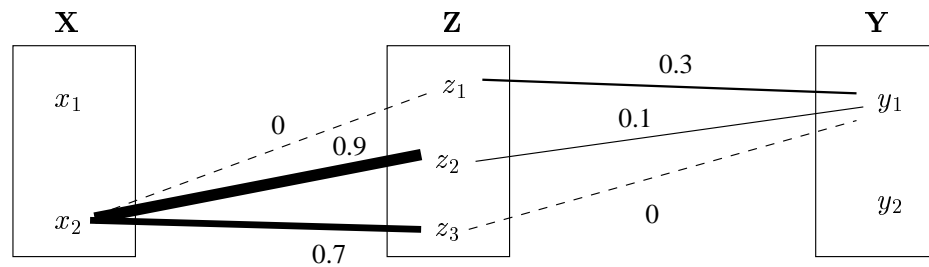


Figura 3.9: Composição R explicitando a composição das variáveis (x_2, y_1) .

O resultado da composição R , considerando-se o *min* como norma-t (representado pelo operador \wedge), é dado por:

$$R = \begin{array}{c} x_1 \\ x_2 \end{array} \begin{array}{cc} y_1 & y_2 \\ \begin{bmatrix} 0.2 & 0.5 \\ \boxed{0.1} & 0.7 \end{bmatrix} \end{array} \quad \begin{array}{l} \mu_R(x_2, y_1) = \sup_z [0 \wedge 0.3, 0.9 \wedge 0.1, 0.7 \wedge 0] ; \\ \mu_R(x_2, y_1) = \sup_z [0, 0.1, 0] = 0.1 . \end{array}$$

- composição inf-s: $R = G \bullet W$

$$\mu_R(x, y) = \inf_{z \in Z} [\mu_G(x, z) \mathbf{s} \mu_W(z, y)] .$$

O exemplo mais comum é a composição *min-max*.

3.2 Computação com Regras Nebulosas: Raciocínio Aproximado

As regras nebulosas são ferramentas eficientes para a modelagem de sentenças em linguagem natural ou artificial (Jang *et al.*, 1997). Esta modelagem depende, entre outras coisas, do conceito de variável lingüística. A interpretação das regras nebulosas como relações nebulosas apropriadas permite a investigação de diferentes esquemas de raciocínio aproximado. Nestes esquemas, procedimentos de inferência baseados no conceito de regra composicional de inferência são utilizados para derivar conclusões, a partir de um conjunto de regras nebulosas e fatos conhecidos. Nesta seção, serão apresentados os conceitos de variáveis lingüísticas que, juntamente com a definição da regra composicional de inferência, permite a ligação entre as regras nebulosas e o raciocínio aproximado.

3.2.1 Variáveis Lingüísticas

Uma variável lingüística pode ser definida, de uma maneira informal, como uma variável cujos valores são palavras ou sentenças, ao invés de números (Pedrycz & Gomide, 1998). A cobertura do domínio da variável com uma série de conjuntos nebulosos e a definição do significado de cada conjunto resultam no conceito formal de variável lingüística (Berthold & Hand, 1999). Para Zadeh (1975), uma variável lingüística é dada por uma quintupla:

$$\langle X, \mathcal{T}(X), \mathbf{X}, \mathcal{G}, \mathcal{M} \rangle, \text{ onde}$$

X → nome da variável lingüística, cuja variável base é x ;

$\mathcal{T}(X)$ → conjunto de termos lingüísticos. Cada elemento de $\mathcal{T}(X)$ representa um rótulo l dos termos que a variável X pode assumir;

\mathbf{X} → universo de discurso da variável lingüística X ;

\mathcal{G} → gramática para a geração dos termos ou rótulos;

\mathcal{M} → regra que associa a cada rótulo (l), um conjunto nebuloso no universo \mathbf{X} , representando o seu significado $\mathcal{M}(l)$. Por exemplo, considere a variável lingüística *velocidade* ($X = \textit{velocidade}$) com universo $\mathbf{X} = [0, 150]$ e variável base $x \in \mathbf{X}$. Um possível conjunto de termos associados à variável *velocidade* poderia ser

$$\mathcal{T}(\textit{velocidade}) = \{\text{muito-baixa, baixa, média, alta, muito-alta}\} .$$

Seja $\mathcal{F}(\mathbf{X})$ uma família de conjuntos nebulosos definidos no universo \mathbf{X} . Então, a regra \mathcal{M} mapeia $\mathcal{T}(X) \rightarrow \mathcal{F}(\mathbf{X})$, ou seja, associa a cada rótulo do conjunto de termos $\mathcal{T}(X)$, um conjunto nebuloso definido em \mathbf{X} .

A gramática \mathcal{G} define como os termos primários {baixa, média, alta} serão associados aos modificadores {muito, pouco, maior, menor, ou, não} para formar os nomes dos termos não-primários. Usualmente, as funções de pertinência associadas às variáveis lingüísticas através de termos primários (baixa, média, alta), têm formatos conhecidos: triangular, trapezoidal, Gaussiana. Os conjuntos nebulosos associados aos termos não-primários (muito-alta, pouco-alta, não-alta) podem ser derivados através do uso de modificadores pré-especificados (para maiores detalhes, ver Pedrycz & Gomide (1998)).

A granularização, i.e. a especificação e distribuição dos termos lingüísticos define a partição nebulosa do universo correspondente. Uma partição nebulosa pode ser uniforme (com todos os termos iguais e equidistantes) ou não-uniforme. O número de termos lingüísticos define a granularidade das partições de cada universo. Um número pequeno de termos lingüísticos define uma partição esparsa ou grossa, ao passo que um número maior resulta numa partição fina. A figura 3.10 mostra dois exemplos de possíveis partições para a variável lingüística *velocidade*, definida em X .

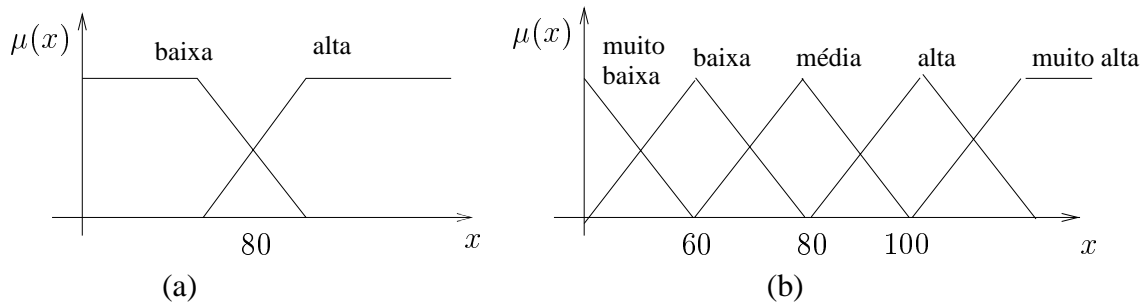


Figura 3.10: Partições uniformes no universo X da variável *velocidade*: (a) partição esparsa, (b) partição fina.

A partição do universo também pode ser vista como um forma de compressão nebulosa de dados. Utilizando agrupamentos nebulosos de informações similares (*fuzzy clusters*), pode-se esconder parte da informação inútil, indesejada ou redundante. Assim, a granularização da partição pode ser usada para focar a análise nos aspectos de interesse do usuário (Berthold & Hand, 1999).

Em grande parte das vezes, a partição nebulosa é obtida manualmente, através da intervenção de um especialista. No entanto, cresce a cada dia o número de abordagens automáticas que definem, através de dados de entrada-saída, a melhor partição para a aplicação em questão. Existem abordagens automáticas baseadas em métodos de agrupamento, como o proposto por Simpson (1992) e Castellano & Fanelli (2000), Rizzi *et al.* (1999). Outras abordagens utilizam algoritmos genéticos como, por exemplo, o método proposto por Cordon *et al.* (2001a) para aplicações envolvendo um grande número de variáveis; ou a abordagem baseada em evolução, proposta por Delgado *et al.* (2001a), que apresenta um método hierárquico-evolutivo para o ajuste da melhor partição do universo de discurso. Na ausência do especialista e de abordagens para o ajuste automático, as partições contendo de 5 a 7 termos lingüísticos, com os conjuntos nebulosos uniformemente distribuídos ao longo do universo, são comumente utilizadas (Pedrycz & Gomide, 1998).

3.2.2 Regras Nebulosas

As regras nebulosas, também conhecidas como implicações nebulosas ou declarações condicionais nebulosas, permitem uma maneira formal de representação de diretivas e estratégias. As regras nebulosas são muito apropriadas quando o conhecimento do domínio resulta de associações empíricas e experiências do operador humano, ou quando se deseja uma representação lingüística do conhecimento adquirido.

Em geral, as regras nebulosas assumem a forma “se *<antecedente>* então *<conseqüente>*”, na qual os antecedentes e os conseqüentes são representados por proposições P_a e P_c , respectivamente. Em casos mais simples, o antecedente P_a e o conseqüente P_c podem ser definidos como proposições atômicas que representam unidades de informação básica, e são dadas por $P_a : X \text{ é } A$ e $P_c : Y \text{ é } B$, resultando em regras na forma:

Se $X \text{ é } A$ então $Y \text{ é } B$,

onde A e B representam os termos lingüísticos e são definidos por conjuntos nebulosos nos universos \mathbf{X} e \mathbf{Y} , respectivamente. A expressão “se *velocidade é alta* então *pressão é baixa*” é um exemplo de uma regra nebulosa simples que relaciona as variáveis lingüísticas *velocidade* e *pressão*, combinando os conjuntos nebulosos associados aos termos lingüísticos *alta* e *baixa*.

Toda regra nebulosa do tipo “se $X \text{ é } A$ então $Y \text{ é } B$ ”, abreviada por $A \rightarrow B$, onde A e B representam conjuntos nebulosos, pode ser definida como uma relação nebulosa R no produto cartesiano $\mathbf{X} \times \mathbf{Y}$. Seja f uma função na forma $f : [0, 1]^2 \rightarrow [0, 1]$, que especifica a semântica da regra $A \rightarrow B$. De uma forma geral, R pode ser descrita por uma função de pertinência, definida no espaço bidimensional como:

$$\mu_R(x, y) = f(\mu_A(x), \mu_B(y)) = f(a, b),$$

onde $a = \mu_A(x)$, $b = \mu_B(y)$.

Segundo Pedrycz & Gomide (1998), as relações nebulosas induzidas por regras nebulosas são derivadas de três classes principais de funções f : conjunções nebulosas (f_t), disjunções nebulosas (f_s) e implicações nebulosas (f_i), sendo as conjunções e as implicações as mais comuns. As conjunções e as disjunções nebulosas podem ser vistas como generalizações duais do produto cartesiano nebuloso através de normas-t e normas-s, enquanto que as implicações nebulosas representam generalizações das implicações da lógica multi-valores de Lukasiewicz (1970).

- Conjunção nebulosa: A conjunção nebulosa representando a relação $R : A \rightarrow B = A \times B$ é

uma função $f_t : [0, 1]^2 \rightarrow [0, 1]$ definida por

$$f_t(\mu_A(x), \mu_B(y)) = \mu_A(x) \mathbf{t} \mu_B(y) .$$

Algumas das normas-t descritas na seção 3.1.3 podem ser empregadas para se definir uma conjunção nebulosa. As definições mais comuns são as de Mamdani (Mamdani & Assilian, 1975) e Larsen (Larsen, 1980), que utilizam o mínimo e o produto algébrico, respectivamente. Seja o operador min representado por \wedge . Então:

$$\text{Mamdani: } f_m(\mu_A(x), \mu_B(y)) = \mu_A(x) \wedge \mu_B(y) ;$$

$$\text{Larsen: } f_p(\mu_A(x), \mu_B(y)) = \mu_A(x)\mu_B(y) .$$

- Disjunção nebulosa: A disjunção nebulosa representando a relação $R : A \rightarrow B = A \times B$ é uma função $f_s : [0, 1]^2 \rightarrow [0, 1]$ definida por

$$f_s(\mu_A(x), \mu_B(y)) = \mu_A(x) \mathbf{s} \mu_B(y) ;$$

- Implicação Nebulosa: A implicação nebulosa representando a relação $R : A \rightarrow B = A \Rightarrow B$ é uma função $f_i : [0, 1]^2 \rightarrow [0, 1]$ que pode ser classificada em duas categorias básicas:

- Implicação-S: $f_{iS}(\mu_A(x), \mu_B(y)) = \mu_{\bar{A}}(x) \mathbf{s} \mu_B(y) .$

Esta categoria de implicação resulta do formalismo da lógica clássica

$$p \Rightarrow q = \bar{p} \vee q .$$

- Implicação-R: $f_{iR}(\mu_A(x), \mu_B(y)) = \sup_{c \in [0,1]} \{ \mu_A(x) \mathbf{t} c \leq \mu_B(y) \}, \forall (x, y) \in \mathbf{X} \times \mathbf{Y} .$

O exemplo mais conhecido de implicação nebulosa é a implicação de Lukasiewicz:

$$f_l(\mu_A(x), \mu_B(y)) = \min [1, 1 - \mu_A(x) + \mu_B(y)] .$$

A figura 3.11 ilustra as duas interpretações mais comuns (conjunção e implicação, representadas pelas regiões em cinza) para a relação ou regra nebulosa $A \rightarrow B$.

As regras do tipo “Se X é A então Y é B ” representam regras nebulosas com proposições monovariáveis tanto no antecedente quanto no conseqüente. Na maioria das aplicações práticas, no entanto, proposições mais complexas envolvendo mais de uma variável (proposições multivariáveis) são utilizadas. As regras nebulosas do tipo “se P_a então P_c ” com proposições multivariáveis,

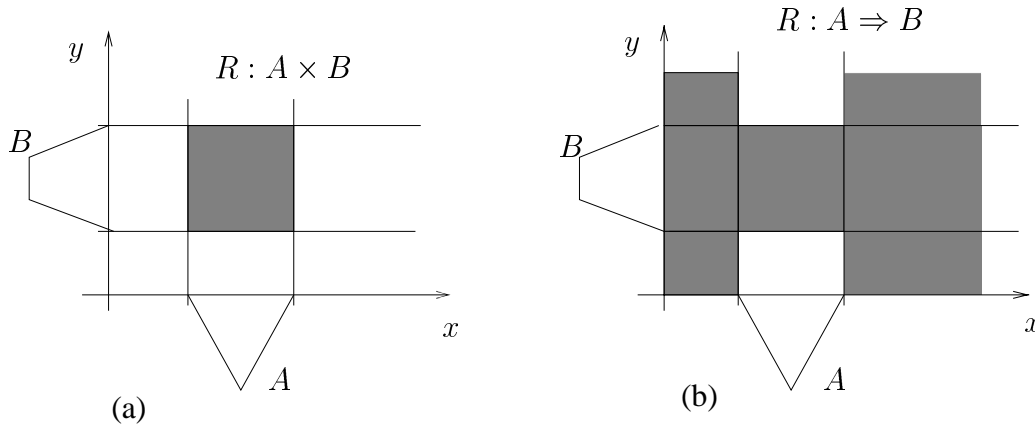


Figura 3.11: Interpretação para a regra “Se A então B ”: (a) conjunção f_t (b) implicação f_i .

são representadas por declarações nas quais o antecedente P_a e o conseqüente P_c são dados por conjunções do tipo:

$$P_a : X_1 \acute{e} A_1 \text{ E } X_2 \acute{e} A_2 \text{ E } \cdots \text{ E } X_n \acute{e} A_n ;$$

$$P_c : Y_1 \acute{e} B_1 \text{ E } Y_2 \acute{e} B_2 \text{ E } \cdots \text{ E } Y_s \acute{e} B_s .$$

onde X_1, X_2, \dots, X_n são variáveis lingüísticas do antecedente e Y_1, Y_2, \dots, Y_s representam as variáveis do conseqüente; e A_1, A_2, \dots, A_n e B_1, B_2, \dots, B_s são conjuntos nebulosos nos universos $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n$ e $\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_s$, respectivamente.

Proposições do tipo disjuntivas, embora menos comuns, também podem ser utilizadas. Nestes casos, para regras do tipo “se Q_a então Q_c ”, Q_a e Q_c são dados por:

$$Q_a : X_1 \acute{e} A_1 \text{ OU } X_2 \acute{e} A_2 \text{ OU } \cdots \text{ OU } X_n \acute{e} A_n ;$$

$$Q_c : Y_1 \acute{e} B_1 \text{ OU } Y_2 \acute{e} B_2 \text{ OU } \cdots \text{ OU } Y_s \acute{e} B_s .$$

As proposições multivariáveis induzem relações nebulosas P_a, Q_a em $\mathbf{X}_1 \times \mathbf{X}_2 \cdots \times \mathbf{X}_n$ e P_c, Q_c em $\mathbf{Y}_1 \times \mathbf{Y}_2 \cdots \times \mathbf{Y}_s$ tais que

$$P_a(x_1, \dots, x_n) = Q_a(x_1, \dots, x_n) = \mathbf{A}_a [\mu_{A_k}(x_k)] ;$$

$$P_c(y_1, \dots, y_s) = Q_c(y_1, \dots, y_s) = \mathbf{A}_c [\mu_{B_k}(y_k)] .$$

onde \mathbf{A}_a e \mathbf{A}_c são normas-t para as relações P_a e P_c e normas-s para as relações Q_a e Q_c , aplicadas a uma série de conjuntos nebulosos, vistas como operadores de agregação (conjunção) tanto dos antecedentes quanto dos conseqüentes.

Alguns sistemas nebulosos aceitam regras com condições irrelevantes (*don't care conditions*) nos antecedentes e/ou nos conseqüentes. Para estas regras, denominadas regras genéricas, nem todas as variáveis disponíveis são relevantes para a obtenção da saída inferida por aquela regra. Por exemplo, supondo uma regra do tipo “Se P_a então P_c ”, e a irrelevância da primeira variável no antecedente e da última variável do conseqüente, as proposições conjuntivas P_a e P_c seriam dadas por:

$$P_a : X_1 \text{ é } \# \text{ E } X_2 \text{ é } A_2 \text{ E } \dots \text{ E } X_n \text{ é } A_n$$

$$P_c : Y_1 \text{ é } B_1 \text{ E } Y_2 \text{ é } B_2 \text{ E } \dots \text{ E } Y_s \text{ é } B_{s-1} \text{ E } Y_s \text{ é } \# .$$

onde os símbolos $\#$ indicam a irrelevância das variáveis correspondentes, cujos conjuntos nebulosos são representados pela função de pertinência de irrelevância descrita na seção 3.1.1. Deste modo, proposições envolvendo um grande número de variáveis podem ser substituídas por proposições mais simples, sem a presença das variáveis irrelevantes. Para o exemplo mostrado anteriormente, as proposições P_a e P_c seriam equivalentes e poderiam ser substituídas pelas proposições P'_a e P'_c na forma:

$$P'_a : X_2 \text{ é } A_2 \text{ E } \dots \text{ E } X_n \text{ é } A_n ;$$

$$P'_c : Y_1 \text{ é } B_1 \text{ E } Y_2 \text{ é } B_2 \text{ E } \dots \text{ E } Y_{s-1} \text{ é } B_{s-1} .$$

3.2.3 Raciocínio Aproximado

Todo método de raciocínio pode ser definido como um processo de inferência que produz conclusões a partir de um conjunto formado por uma ou mais regras e fatos conhecidos.

Na lógica bi-valores tradicional, o *modus ponens* (MP) é a regra básica de inferência utilizada:

$$\begin{array}{l} \text{(fato):} \quad X \text{ é } A \\ \text{Modus Ponens} \quad \text{(regra):} \quad \text{se } X \text{ é } A \text{ então } Y \text{ é } B \\ \hline \text{(conclusão):} \quad Y \text{ é } B . \end{array}$$

Como exemplo, considere o fato de que o “tomate é vermelho” e a regra “se o tomate é vermelho então ele está maduro”. Pode-se então concluir que “o tomate está maduro”.

O raciocínio aproximado envolve a computação com regras imprecisas e mecanismos de inferência específicos. Conforme discutido por Jang *et al.* (1997), no raciocínio humano tem-se o *modus ponens* utilizado de maneira aproximada, resultando no *modus ponens* generalizado:

$$\begin{array}{l} \text{(fato):} \quad X \text{ é } A' \\ \text{Modus Ponens Generalizado} \quad \text{(regra):} \quad \text{se } X \text{ é } A \text{ então } Y \text{ é } B \\ \hline \text{(conclusão):} \quad Y \text{ é } B' . \end{array}$$

Neste caso, tem-se o fato de que “tomate é mais ou menos vermelho”, podendo concluir-se que “o tomate está mais ou menos maduro”.

Quando A , A' , B e B' são conjuntos nebulosos, o método de raciocínio é chamado de raciocínio nebuloso. Por ser uma forma de raciocínio aproximado e utilizar o *modus ponens* generalizado para a computação com regras nebulosas, muitas vezes estes termos são utilizados como sinônimos do raciocínio nebuloso. No entendimento do raciocínio nebuloso, o primeiro passo depende da definição da semântica da regra nebulosa definida na seção 3.2.2. O passo seguinte determina como a conclusão será extraída a partir do fato e da regra nebulosa. Este procedimento, chamado de regra composicional de inferência será discutido na seção a seguir. A regra composicional tem o *modus ponens* generalizado como caso particular.

3.2.4 Regra Composicional de Inferência

O procedimento conhecido como regra composicional de inferência, ou simplesmente regra da composição, pode ser visto como uma generalização do processo de se inferir o valor b (ou valores do intervalo \tilde{b}) de uma função $f(\cdot)$, a partir de um ponto $x = a$ (ou intervalo $x = \tilde{a}$) qualquer no universo X . Este mecanismo consiste em se encontrar o ponto $b = f(a)$ (ou intervalo $\tilde{b} = f(\tilde{a})$). Para isto constrói-se uma reta (ou retas) paralela(s) ao eixo Y , a partir do ponto $x = a$ (ou intervalo $x = \tilde{a}$), chamada(s) de extensão cilíndrica $cil(a)$ (ou $cil(\tilde{a})$) com a curva f . A projeção de I no eixo Y resulta no ponto $y = b$ (ou intervalo $y = \tilde{b}$). A figura 3.12 ilustra esta idéia.

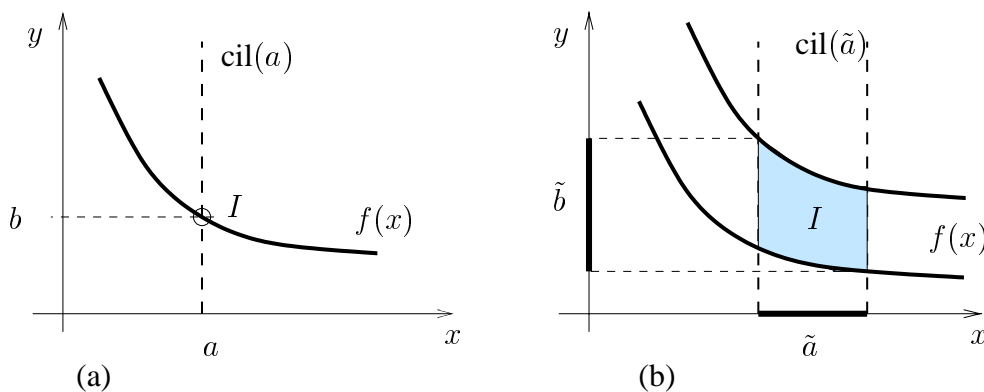


Figura 3.12: Processo de inferência com a função $f(x)$: (a) ponto-a-ponto, (b) intervalo.

Para a formalização da regra composicional de inferência, envolvendo dois conjuntos nebulosos A' e B' e a relação nebulosa R , é necessária a definição dos conceitos de extensão cilíndrica e

projeção para conjuntos nebulosos.

- **extensão cilíndrica:** a extensão cilíndrica em $\mathbf{X} \times \mathbf{Y}$ de qualquer conjunto nebuloso A em \mathbf{X} é uma relação nebulosa com função de pertinência

$$\mu_{\text{cil}(A)}(x, y) = \mu_A(x) ;$$

- **projeção:** a projeção no eixo \mathbf{X} de uma relação nebulosa R em $\mathbf{X} \times \mathbf{Y}$ é dada por um conjunto nebuloso com função de pertinência

$$\mu_{\text{proj}_{\mathbf{X}}(R)}(x) = \sup_{y \in \mathbf{Y}} [\mu_R(x, y)] .$$

A extensão cilíndrica e a projeção têm comportamentos complementares, pois afetam as dimensões das relações ou conjuntos nebulosos envolvidos de forma diferente. A extensão cilíndrica eleva a dimensão do argumento expandindo, por exemplo, um conjunto nebuloso para uma relação nebulosa, uma relação nebulosa bidimensional para uma relação nebulosa tridimensional, e assim por diante. A projeção faz o caminho inverso, reduzindo a dimensão do objeto (Pedrycz & Gomide, 1998).

A idéia da extensão cilíndrica é obter objetos compatíveis em termos de dimensão. Por exemplo, o processo composicional de inferência depende da composição de um conjunto nebuloso A em \mathbf{X} com uma relação nebulosa R em $\mathbf{X} \times \mathbf{Y}$. Entretanto, as dimensões de A e R são incompatíveis. Para se obter a compatibilidade necessária, a extensão cilíndrica é utilizada para aumentar a dimensão do conjunto A .

Uma vez definida a extensão cilíndrica e projeção no âmbito de conjuntos nebulosos, é possível se extrair um conjunto nebuloso B' a partir da composição da extensão cilíndrica $\text{cil}(A')$ com a relação nebulosa R , ambas definidas em $\mathbf{X} \times \mathbf{Y}$. Este procedimento, denominado de regra composicional de inferência, foi introduzido por Zadeh (1975) e compreende os seguintes passos:

1. Obter a extensão cilíndrica $\text{cil}(A')$;
2. Encontrar a interseção I entre a extensão cilíndrica $\text{cil}(A')$ e a relação nebulosa R
3. Projetar I em \mathbf{Y} para obter B' .

O conjunto nebuloso B' é obtido projetando-se $I = \text{cil}(A') \cap R$ no eixo \mathbf{Y} , ou seja,

$$B' = \text{proj}_{\mathbf{Y}}(I) = \text{proj}_{\mathbf{Y}}(\text{cil}(A') \cap R) .$$

Considerando-se que a interseção pode ser realizada por qualquer norma-t e a projeção é definida pelo operador sup, a função de pertinência de B' é dada por:

$$\begin{aligned}\mu_{B'}(y) &= \sup_x [\mu_{\text{cil}(A')}(x, y) \mathbf{t} \mu_R(x, y)] \\ &= \sup_x [\mu_{A'}(x) \mathbf{t} \mu_R(x, y)] ,\end{aligned}$$

o que resulta numa composição sup-t. Portanto,

$$B' = A' \circ R .$$

O princípio da extensão é bastante utilizado em teoria da informação (Pedrycz & Gomide, 1998) e generaliza mapeamentos ponto-a-ponto de funções, para mapeamentos entre conjuntos nebulosos. Conforme apontado por Jang *et al.* (1997), o princípio da extensão é apenas um caso especial da regra composicional de inferência, no qual a relação nebulosa R é substituída por uma função $f(\cdot)$. A figura 3.13 ilustra o mapeamento ponto-a-ponto, o princípio da extensão e o mecanismo da regra da composição, onde a região em cinza representa uma relação nebulosa R qualquer.

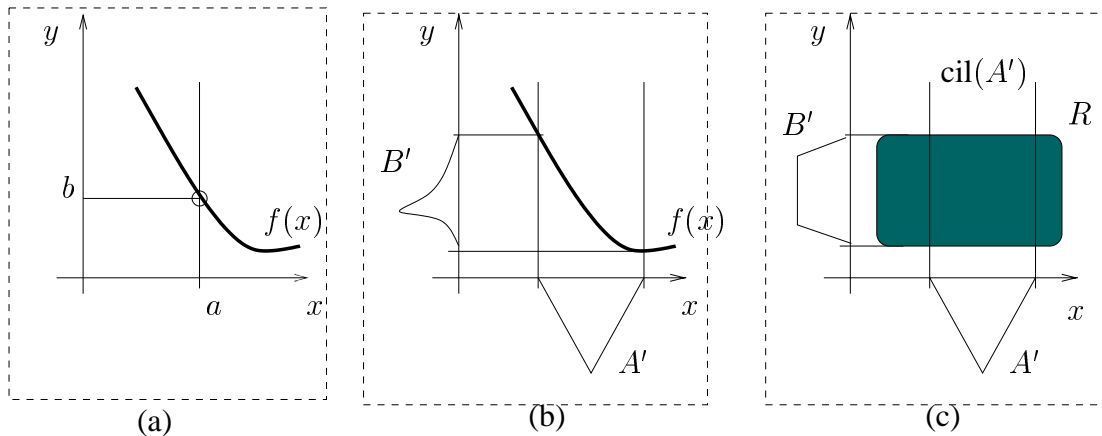


Figura 3.13: Processos de Inferência (a) ponto-a-ponto; (b) princípio da extensão (c) regra composicional de inferência: $B' = A' \circ R$.

3.2.5 O Processo de Inferência no Raciocínio Aproximado

Utilizando-se a regra composicional de inferência, é possível definir um procedimento para se obter uma conclusão nebulosa B' a partir de uma ou mais regras (representadas pela relação nebulosa R) e de um fato A' . Para isso é necessária a definição da relação R que representa a base de regras nebulosas.

Por simplicidade, supõe-se o raciocínio nebuloso envolvendo apenas duas variáveis lingüísticas X e Y , definidas nos universos \mathbf{X} e \mathbf{Y} , respectivamente. A extensão para mais variáveis é obtida de forma direta pela utilização de regras com proposições multivariáveis definidas na seção 3.2.2. Estes mecanismos de inferência mais complexos serão detalhados na seção 3.3, que traz algoritmos para a inferência em sistemas nebulosos.

Sejam A' e B' conjuntos nebulosos em \mathbf{X} e \mathbf{Y} , respectivamente. Supondo-se uma base de regras representada pela relação nebulosa R definida em $\mathbf{X} \times \mathbf{Y}$, o conjunto nebuloso B' , induzido pelo fato $X \text{ é } A'$ e pela relação R , é dado por:

Raciocínio Nebuloso	(fato):	$X \text{ é } A'$
	(regras):	R
	(conclusão):	$Y \text{ é } B' = A' \circ R$.

No raciocínio nebuloso, R pode representar uma regra individual ou um conjunto de regras. Para uma única regra $R : A \rightarrow B$, o processo de inferência de B' a partir do fato A' e da regra nebulosa R , seria dado por

$$B' = A' \circ R = A' \circ (A \rightarrow B).$$

Supondo-se A , B e A' conjuntos nebulosos quaisquer em \mathbf{X} , \mathbf{Y} e \mathbf{X} , respectivamente, a figura 3.14 ilustra o processo de inferência, no qual a semântica da regra R é dada por duas funções diferentes (conjunção e implicação representadas pela região em cinza). No caso específico da conjunção nebulosa, a relação $R : A \rightarrow B = A \times B$ pode ser vista como um grânulo (ponto nebuloso).

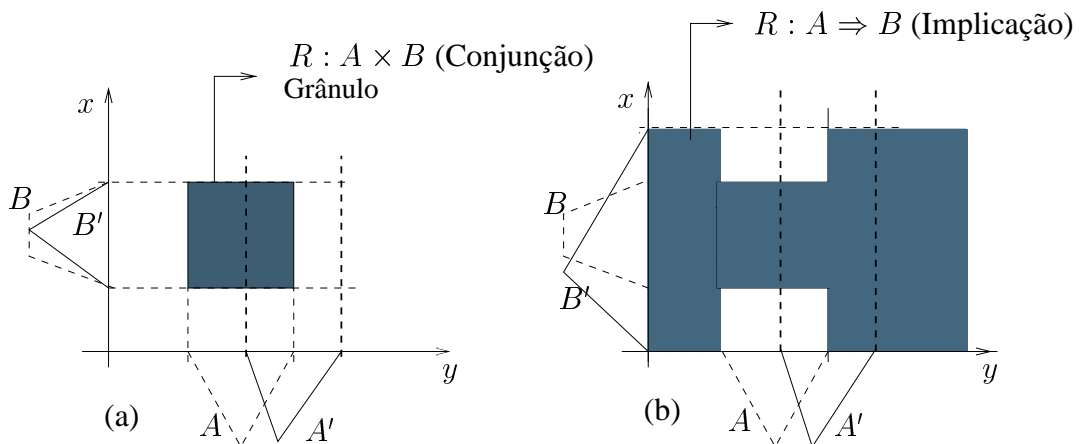


Figura 3.14: Representação no plano do raciocínio nebuloso envolvendo uma única regra: (a) Regra dada pela conjunção nebulosa $R : A \times B$, (b) Regra dada pela implicação nebulosa $R : A \Rightarrow B$.

A representação no plano, não detalha como foi obtido o formato final do conjunto nebuloso inferido. Os formatos da conclusão, assim como todas as etapas do processo de inferência presentes no raciocínio nebuloso ficam melhor ilustradas numa representação espacial, como mostrado na figura 3.15. A figura detalha o mecanismo do raciocínio nebuloso $B' = A' \circ R$ através da regra composicional de inferência. Aqui, supõe-se que a semântica da regra é dada por Larsen, ou seja, $R : A \rightarrow B = f_p(\mu_A(x), \mu_B(y)) = \mu_A(x)\mu_B(y)$. A extensão cilíndrica é aplicada a o conjunto A' para tornar as suas dimensões compatíveis com a relação R . A interseção $I = \text{cil}(A') \cap R$, associada à norma-t da composição sup-t, é dada pelo operador *min*, isto é, $\mu_I(x, y) = \min[\mu_{A'}(x, y), \mu_R(x, y)]$. Por fim, o conjunto B' é obtido pela projeção da interseção I no eixo Y .

No caso de mais de uma regra, o raciocínio nebuloso geralmente é realizado sobre um conjunto de m regras nebulosas na forma:

$$R_j: \text{Se } X \text{ é } A^j \text{ então } Y \text{ é } B^j, j = 1, \dots, m.$$

Cada regra individual R_j pode induzir uma relação nebulosa diferente. O conjunto de regras, por sua vez, resulta numa relação nebulosa que é obtida pela agregação das relações (ou regras) individuais, ou seja,

$$R = \mathbf{A}_R (R_j) = \mathbf{A}_R (A^j \rightarrow B^j) .$$

O operador de agregação \mathbf{A}_R é normalmente caracterizado por uma norma-s, mas normas-t e operadores de média também podem ser utilizados. O processo de inferência é ilustrado na figura 3.16. No caso da semântica de cada regra ser dada por uma conjunção nebulosa, um conjunto de grânulos forma um grafo nebuloso.

3.3 Sistemas Nebulosos

Os sistemas nebulosos, também conhecidos como sistemas de inferência nebulosa, sistemas nebulosos baseados em regras, ou modelos nebulosos, representam a mais importante ferramenta de modelagem baseada na teoria dos conjuntos nebulosos. Os sistemas nebulosos têm sido aplicados com sucesso em diversas áreas, como: controle automático, classificação e reconhecimento de padrões, tomada de decisão, sistemas inteligentes, previsão de séries temporais e robótica (Jang *et al.*, 1997).

Conforme discutido por Lin & Lee (1996), um sistema de inferência nebulosa é um mapeamento ou função de um espaço de alternativas de entrada para um espaço de saída. Segundo Jang *et al.*

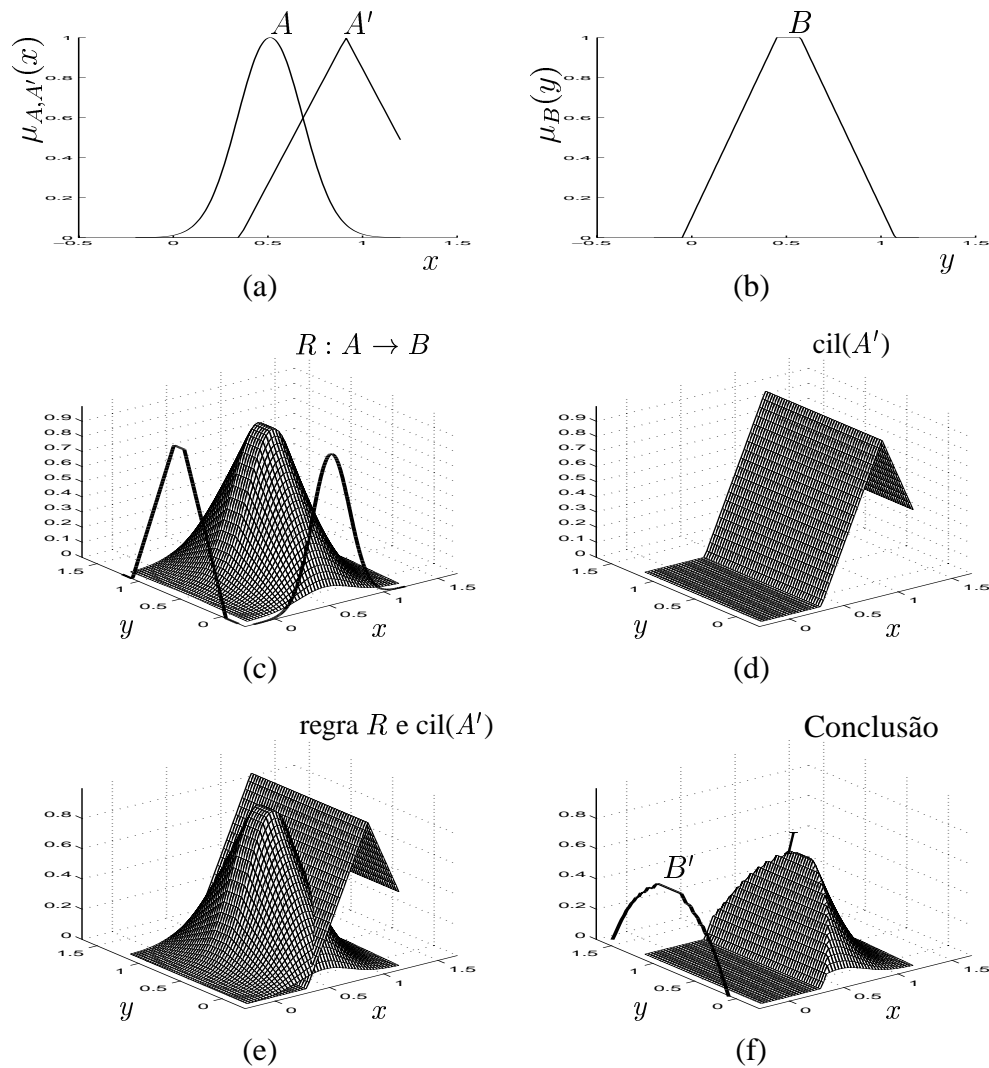


Figura 3.15: Representação no espaço do raciocínio aproximado envolvendo uma única regra: (a) Conjuntos Nebulosos A e A' , (b) Conjunto Nebuloso B , (c) Regra Nebulosa $A \rightarrow B$ dada pelo produto algébrico, (d) Extensão Cilíndrica de A' , (e) Regra Nebulosa e Extensão Cilíndrica, (f) Interseção I e projeção para obtenção da conclusão B' .

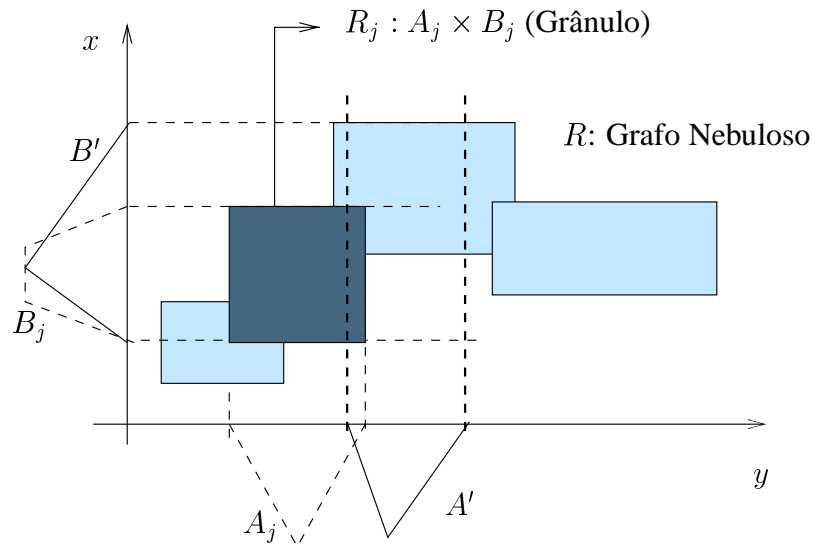


Figura 3.16: Representação no plano do raciocínio nebuloso envolvendo uma base de regras que define um grafo nebuloso.

(1997), a estrutura básica de um sistema nebuloso possui três componentes conceituais: uma **base de regras**, que contém o conjunto de regras nebulosas; uma **base de dados**, que define as funções de pertinência usadas nas regras nebulosas; e um **mecanismo de raciocínio**, que realiza um procedimento de inferência (raciocínio nebuloso) para obter a saída ou conclusão, baseado nas regras e fatos conhecidos. Neste trabalho, a maioria dos parâmetros relacionados a estes componentes conceituais será obtido de forma automática pela abordagem que está sendo proposta.

3.3.1 Métodos de Transformação da Saída

Em muitas situações práticas, é necessário que se aplique uma transformação na saída inferida pelo sistema nebuloso. Em problemas de controle, por exemplo, existe a necessidade de uma saída não-nebulosa (*crisp*). Em modelos que utilizam conseqüentes lingüísticos, como é o caso dos modelos Mamdani, a obtenção de uma saída *crisp* requer um estágio de transformação da informação nebulosa em informação não-nebulosa (*defuzzification*). Existem diferentes métodos que implementam esta transformação. Dentre os mais utilizados destacam-se: média dos máximos, centro de massa e centro de área. Seja C um conjunto nebuloso definido no universo Z , então:

- Média dos Máximos (MoM): Os valores do domínio correspondentes ao máximo da função de pertinência do conjunto C são identificados e a média define o valor não-nebuloso;

- Centro de Massa (CoG): O resultado da transformação de C em um valor não-nebuloso é o centro de massa, dado por:

$$\hat{z} = \frac{\int_z \mu_C(z) z dz}{\int_z \mu_C(z) dz};$$

- Centro de Área (CoA): Neste caso, \hat{z} resulta do balanço entre duas áreas de C determinada por \hat{z} :

$$\int_{-\infty}^{\hat{z}} \mu_C(z) dz = \int_{\hat{z}}^{\infty} \mu_C(z) dz .$$

Em alguns modelos, a saída obtida de cada regra é não-nebulosa (*crisp*) mas, para aumentar a interpretabilidade do modelo, a aplicação poderia exigir uma informação lingüística no conseqüente de cada regra. Neste sentido, algumas abordagens têm proposto métodos para a transformação de informação não-nebulosa (*crisp*) em informação nebulosa, representada por funções de pertinência associadas a termos lingüísticos específicos. Para maiores detalhes ver (Espinosa & Vandewalle, 2000; Setnes *et al.*, 1998).

3.3.2 Mecanismos de Inferência

Os sistemas nebulosos utilizam uma computação baseada na regra composicional de inferência descrita na seção 3.2.4 e no raciocínio nebuloso mostrado na seção 3.2.5. De uma forma geral, tem-se:

$$\begin{array}{ll} \text{(fato } P_i): & X_1 \text{ é } A_1 \text{ E } X_2 \text{ é } A_2 \text{ E } \dots \text{ E } X_n \text{ é } A_n \\ \text{(regra } R_1): & \text{Se } X_1 \text{ é } A_1^1 \text{ E } \dots \text{ E } X_n \text{ é } A_n^1 \text{ então } Y_1 \text{ é } B_1^1 \text{ E } \dots \text{ E } Y_s \text{ é } B_s^1 \\ & \vdots \\ \text{(regra } R_m): & \text{Se } X_1 \text{ é } A_1^m \text{ E } \dots \text{ E } X_n \text{ é } A_n^m \text{ então } Y_1 \text{ é } B_1^m \text{ E } \dots \text{ E } Y_s \text{ é } B_s^m \\ \hline \text{(conclusão } P_o): & Y_1 \text{ é } B_1 \text{ E } Y_2 \text{ é } B_2 \text{ E } \dots \text{ E } Y_s \text{ é } B_s . \end{array}$$

A forma como a conclusão é extraída a partir do fato P_i e das regras, R_1 a R_m define diferentes mecanismos de inferência utilizados por sistemas nebulosos. A seguir, serão mostrados os tipos mais comuns.

Inferência Composicional

Segundo Pedrycz & Gomide (1998), o esquema de inferência composicional para um sistema nebuloso composto por m regras nebulosas R_j , $j = 1, \dots, m$, envolve os seguintes passos:

passo 1 Definição dos operadores apropriados. Esta etapa requer a escolha de:

- 1.1 Operadores de agregação (\mathbf{A}_a e \mathbf{A}_c): norma-t apropriada para representar a conjunção E nos antecedentes e conseqüentes de cada regra e proposição de entrada;
- 1.2 Semântica da regra: uma função f para definir a semântica de cada regra nebulosa R_j , $j = 1, \dots, m$;
- 1.3 uma norma-t apropriada para a composição sup-t;
- 1.4 um operador de agregação \mathbf{A}_R para combinar as m regras R_1 a R_m .

passo 2 Cálculos Iniciais

2.1 Calcular $P_i(x_1, x_2, \dots, x_n) = \mathbf{A}_a [\mu_{A_k}(x_k)]$;

2.2 Para $j = 1, \dots, m$ computar:

$$P_{aj}(x_1, x_2, \dots, x_n) = \mathbf{A}_a [\mu_{A_k^j}(x_k)] ,$$

$$P_{cj}(y_1, y_2, \dots, y_s) = \mathbf{A}_c [\mu_{B_k^j}(y_k)] ,$$

$$R_j(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_s) = f(P_{aj}(x_1, x_2, \dots, x_n), P_{cj}(y_1, y_2, \dots, y_s)) ;$$

passo 3 Computar a relação

$$R(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_s) = \mathbf{A}_R (R_j(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_s)) .$$

passo 4 Calcular a proposição de saída (inferida) P_o como

$$P_o(y_1, y_2, \dots, y_s) = \sup_{x_k, k=1, \dots, n} [P_i(x_1, x_2, \dots, x_n) \mathbf{t} R(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_s)] .$$

No esquema geral de inferência descrito anteriormente, a regra composicional é aplicada depois que a relação nebulosa completa R , associada à base de regras, é encontrada. Isto é representado pela expressão:

$$P_o = P_i \circ R = P_i \circ \left(\mathbf{A}_R (R_j) \right) .$$

De forma alternativa, a regra composicional pode ser aplicada localmente para cada relação R_j e as relações resultantes são agregadas para se obter a proposição inferida, na forma:

$$P_o = \mathbf{A}_R (P_i \circ R_j) .$$

Portanto, os passos 3 e 4 do procedimento anterior seriam modificados para:

passo 3a para $j = 1, \dots, m$ computar a proposição inferida de cada regra como:

$$P_{oj}(y_1, y_2, \dots, y_s) = \sup_{x_k, k=1, \dots, n} [P_i(x_1, x_2, \dots, x_n) \mathbf{t} R_j(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_s)]$$

passo 4a Calcular a proposição de saída (inferida) P_o como

$$P_o(y_1, y_2, \dots, y_s) = \mathbf{A}_R (P_{oj}(y_1, y_2, \dots, y_s)) .$$

Em geral, $P_i \circ \left(\mathbf{A}_R (R_j) \right) \neq \mathbf{A}_R (P_i \circ R_j)$, mas em alguns casos especiais a igualdade é observada. São eles:

- composição sup-t com norma-t contínua ou dada pelo *min*;
- operador de agregação \mathbf{A}_R dado pelo máximo

Inferência Escalonada

Um esquema alternativo ao processo de inferência descrito anteriormente é o método conhecido como inferência escalonada (Pedrycz & Gomide, 1998). Neste esquema, a composição $P_i \circ \left(\mathbf{A}_R (R_j) \right)$ é desmembrada de forma que cada proposição atômica do fato seja combinada com a proposição associada no antecedente da regra. O esquema geral para a inferência escalonada é dado pelos seguintes passos:

Passo 1 Matching: para cada regra, computar o nível de similaridade (*matching*) μ_{m_k} , $k = 1, \dots, n$, entre cada proposição atômica (associada à variável k) do antecedente da regra e a proposição atômica correspondente no fato (ou proposição de entrada P_i). A operação de *matching* pode ser relacionada com a composição sup-t. A escolha do *min* como norma-t define o *matching* como uma medida de possibilidade.

Passo 2 Agregação dos Antecedentes: para cada regra, computar o grau de ativação μ_j através da operação de:

- conjunção para regras com proposições atômicas ligadas por E

$$\mu_j = \mathbf{t}_{k=1}^n \mu_{m_k}, j = 1 \dots, m ;$$

- disjunção para regras com proposições atômicas ligadas por OU

$$\mu_j = \mathbf{S}_{k=1}^n \mu_{m_k}, j = 1 \cdots, m .$$

Passo 3 Derivação da conclusão individual P_{oj} : para cada regra, computar o valor inferido, baseado no resultado da agregação dos antecedentes μ_j e na semântica da regra escolhida.

$$P_{oj}(y) = B^{j*}(y) = f(\mu_j, \mu_{B^j}(y)) .$$

Passo 4 Derivação do resultado final P_o : Computar o valor inferido do conjunto completo de regras através da agregação dos valores inferidos no passo 3:

$$P_o(y) = B(y) = \mathbf{A}_R \left(B^{j*} \right) .$$

Como exemplo, suponha o seguinte esquema de raciocínio nebuloso envolvendo regras nebulosas com duas variáveis lingüísticas X_1 e X_2 no antecedente e uma variável lingüística Y no conseqüente,

$$\begin{array}{ll} \text{(fato } P_i): & X_1 \text{ é } A_1 \text{ E } X_2 \text{ é } A_2 \\ \text{(regra } R_1): & \text{Se } X_1 \text{ é } A_1^1 \text{ E } X_2 \text{ é } A_2^1 \text{ então } Y \text{ é } B^1 \\ & \vdots \\ \text{(regras } R_m): & \text{Se } X_1 \text{ é } A_1^m \text{ E } X_2 \text{ é } A_2^m \text{ então } Y \text{ é } B^m \\ \hline \text{(conclusão } P_o): & Y \text{ é } B . \end{array}$$

A utilização da inferência escalonada na implementação do mecanismo de raciocínio nebuloso descrito anteriormente depende da fixação de alguns parâmetros. Suponha a composição dada pelo operador *sup-min* que resulta na medida de *matching* dada pela possibilidade. Suponha ainda que os outros operadores associados ao processo de inferência tenham sido definidos (pelo usuário ou obtidos de forma automática) como:

- agregação de antecedentes \mathbf{A}_a : norma t_2 com parâmetro $p_t = 1.5$;
- semântica das regras f : $f = f_p =$ (produto algébrico)
- a agregação das regras \mathbf{A}_R : max.

Então:

passo 1 *Matching* na regra R_j :

$$\mu_{m_1^j} = \sup_{x_1} \left[\mu_{A_1}(x_1) \wedge \mu_{A_1^j}(x_1) \right] = \text{Poss}(A_1, A_1^j) ;$$

$$\mu_{m_2^j} = \sup_{x_2} \left[\mu_{A_2}(x_2) \wedge \mu_{A_2^j}(x_2) \right] = \text{Poss}(A_2, A_2^j) .$$

passo 2 Agregação dos Antecedentes:

$$\mu_j = \mu_{m_1^j} \mathbf{t}_2 \mu_{m_2^j} ;$$

passo 3 Derivação individual $P_{oj}(y)$:

$$P_{oj}(y) = B^{j*}(y) = f_p(\mu_j, \mu_{B^j}(y)) = \mu_j \mu_{B^j}(y) ;$$

passo 4 Conclusão final

$$P_o(y) = B(y) = \mathbf{A}_R \left(B^{j*} \right) = \max_{j=1}^m [B^{j1*}, \dots, B^{m1*}] .$$

O esquema de inferência escalonada simplifica muito a computação em sistemas de inferência nebulosa. Por isso, a maioria das aplicações práticas utiliza sistemas nebulosos baseados neste mecanismo, como será o caso das simulações realizadas, cujos resultados estão no capítulo 5.

3.3.3 Modelos de Sistemas Nebulosos

Existem vários modelos de sistemas nebulosos. Na maioria dos casos, o antecedente é formado por proposições lingüísticas e a distinção entre os modelos se dá no conseqüente das regras nebulosas. Entre os modelos mais conhecidos podemos destacar:

- o modelo de Mamdani (Mamdani & Assilian, 1975), que utiliza conjuntos nebulosos também nos conseqüentes das regras nebulosas. A saída final é representada por um conjunto nebuloso resultante da agregação da saída inferida de cada regra. Para se obter uma saída final não nebulosa adota-se um dos métodos de transformação da saída nebulosa em não-nebulosa descritos na seção 3.3.1.
- o modelo de Takagi-Sugeno (Takagi & Sugeno, 1983), no qual o conseqüente é representado por uma função das variáveis de entrada. A saída final é obtida pela média ponderada das saídas inferidas de cada regra. Os coeficientes da ponderação são dados pelos graus de ativação das respectivas regras.

- o modelo de Tsukamoto (Tsukamoto, 1979), que utiliza funções de pertinência monotônicas no conseqüente. Assim como no modelo Takagi-Sugeno, é inferido um valor não-nebuloso induzido pelo nível de ativação da regra. A saída final é obtida por média ponderada das saídas inferidas de cada regra.

A seguir, serão detalhados os dois modelos mais comuns: Mamdani e Takagi-Sugeno.

Modelo de Mamdani

O sistema de inferência nebulosa de Mamdani (Mamdani & Assilian, 1975) emergiu como uma primeira tentativa de se controlar a caldeira de uma máquina a vapor, através de um conjunto de regras de controle baseadas em variáveis lingüísticas, regras estas obtidas de operadores humanos especializados (Jang *et al.*, 1997).

A característica básica do modelo de Mamdani é o fato de utilizar conjuntos nebulosos nos conseqüentes das regras nebulosas. As etapas de cálculo de *matching* e agregação dos antecedentes seguem os passos normais da inferência escalonada. A obtenção da conclusão de cada regra depende da semântica escolhida e dos resultados do *matching* e agregação dos antecedentes. No modelo original proposto por Mamdani & Assilian (1975), a semântica é dada pela função $f = f_m = \min$. Atualmente, admite-se a utilização de outros operadores, como, por exemplo, o produto algébrico $f = f_p$. Da agregação das conclusões inferidas de cada regra, resulta um conjunto nebuloso (normalmente não-convexo). Nos casos em que uma saída não-nebulosa é necessária, utiliza-se um dos métodos descritos na seção 3.3.1.

Suponha um sistema nebuloso do tipo Mamdani, composto por duas regras nebulosas na forma

$$R_j : \text{Se } X_1 \text{ é } A_1^j \text{ E } X_2 \text{ é } A_2^j \text{ então } Y \text{ é } B^j, \quad j = 1, 2,$$

onde X_1 e X_2 são variáveis lingüísticas e A_1^j e A_2^j são conjuntos nebulosos nos universos X_1 e X_2 , respectivamente. A figura 3.17 ilustra o processo de inferência (escalonada) para duas entradas não-nebulosas (representadas por valores em posições arbitrárias u_1 e u_2). Neste exemplo, a composição é do tipo *sup-min*, com *matching* dado pela medida de possibilidade, a agregação dos antecedentes é dada pelo *min*, a semântica das regras é fixada como $f = f_m = \min$ e a união das saídas de cada regra é computada pelo operador *max*.

Modelo de Takagi-Sugeno

O modelo Takagi-Sugeno (Takagi & Sugeno, 1983) foi proposto como resultado de um esforço para se desenvolver, de forma sistemática, uma abordagem para a geração de regras nebulosas a

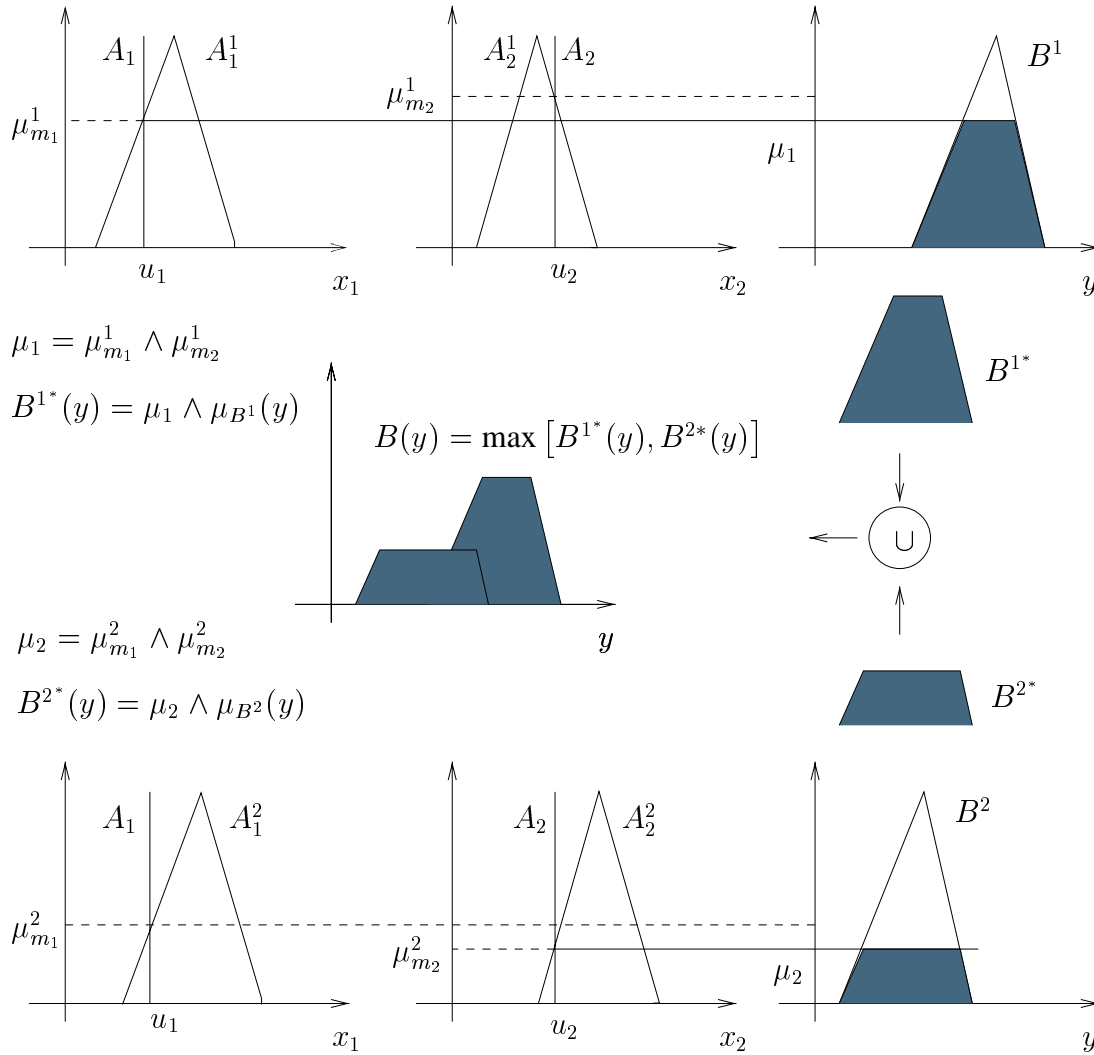


Figura 3.17: Inferência no modelo Mamdani.

partir de dados de entrada-saída. A base de regras nebulosas típica no modelo Takagi-Sugeno (TS), é da forma

$$R_j : \text{Se } X_1 \text{ é } A_1^j \text{ E } X_2 \text{ é } A_2^j \text{ E } \dots \text{ E } X_n \text{ é } A_n^j \text{ então } Y \text{ é } y_j = g_j(\mathbf{w}_j, \mathbf{x}), j = 1..m,$$

onde m define o tamanho da base de regras, X_k , $k = 1, \dots, n$, são variáveis lingüísticas nos universos \mathbf{X}_k , A_k^j , $k = 1, \dots, n$, $j = 1, \dots, m$, são conjuntos nebulosos no antecedente das regras (definidos na base de dados do sistema nebuloso) e $g_j(\mathbf{w}_j, \mathbf{x})$ é um polinômio de saída que é função de dois vetores de parâmetros:

- o vetor $\mathbf{w}_j \in \mathfrak{R}^Q = [w_{1j}, \dots, w_{Qj}]$, representando os parâmetros do conseqüente;

- o vetor $\mathbf{x} \in \mathfrak{R}^n = [x_1, \dots, x_n]$ das variáveis base de entrada.

A ordem do polinômio define a ordem do modelo. Os mais comuns são modelo TS de ordem zero (conseqüentes constantes) e modelo TS de 1ª ordem (conseqüentes lineares). Modelos mais gerais têm sido propostos, como por exemplo o modelo com conseqüentes não-lineares apresentado por Delgado *et al.* (2000a), que será detalhado no capítulo 4.

A figura 3.18 ilustra o mecanismo de inferência (escalonada) utilizado no modelo TS de 1ª ordem com duas regras nebulosas, considerando-se dois valores (u_1 e u_2) na proposição de entrada, e agregação dos antecedentes dada pelo produto algébrico.

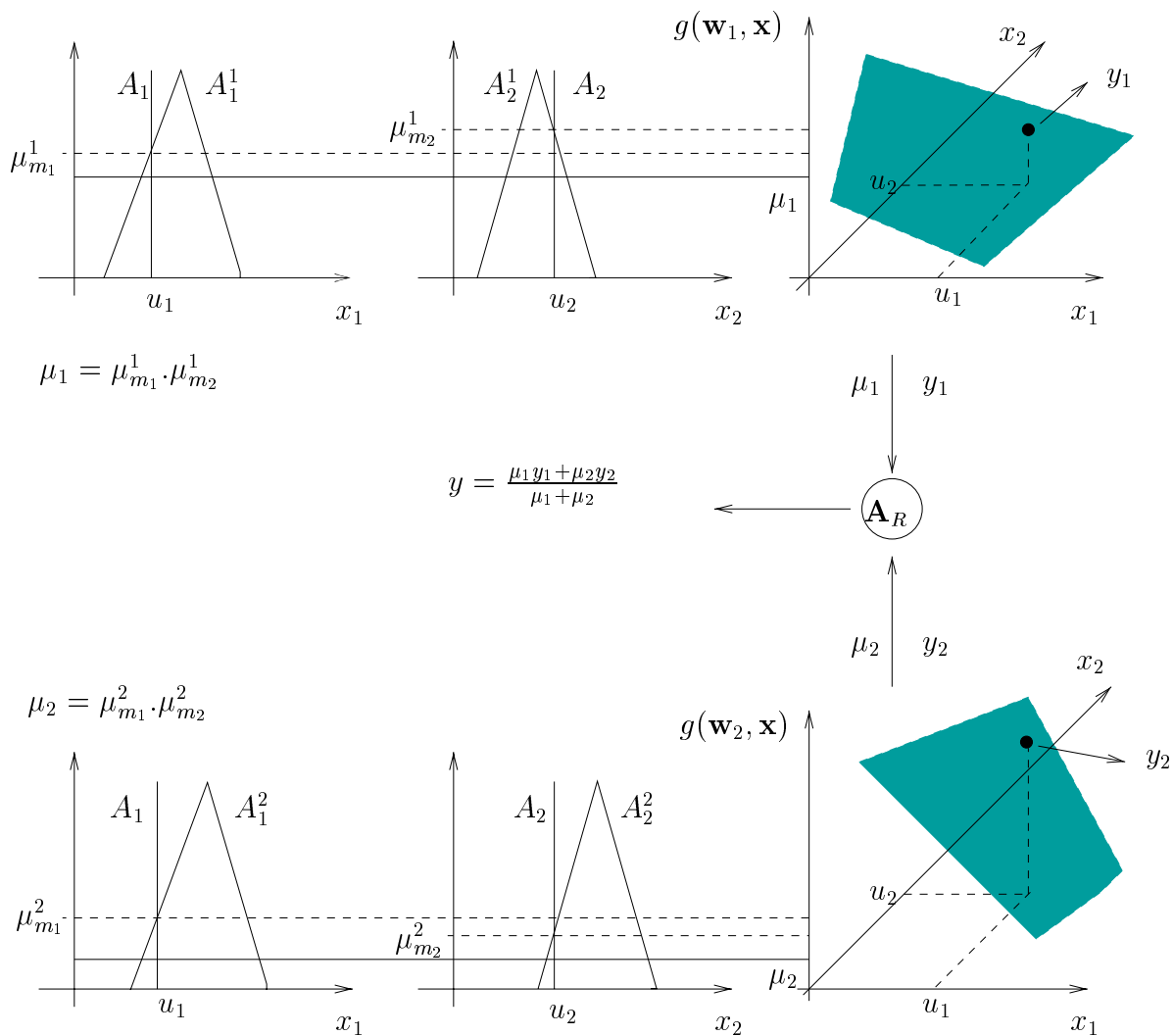


Figura 3.18: Inferência no modelo Takagi-Sugeno.

Uma vez que cada regra tem saída não-nebulosa, a saída final é obtida pela média ponderada dessas saídas e não existe a necessidade da transformação da saída nebulosa em não-nebulosa, requerida pelo modelo de Mamdani. Conforme discutido por Jang *et al.* (1997), esta simplificação, embora seja interessante por exemplo em aplicações de controle com tempo real, pode levar a uma perda do significado lingüístico associado aos sistemas nebulosos. Esta falta de interpretação lingüística no conseqüente e a ausência de conhecimento para a definição dos parâmetros w_j do conseqüente de cada regra fizeram com que os modelos TS fossem preteridos em uma série de aplicações. Nos últimos tempos, a possibilidade de obtenção dos parâmetros dos conseqüentes através de métodos de otimização tem reavivado o interesse por estes modelos, uma vez que os resultados obtidos justificam, em muitas aplicações, a perda de interpretabilidade lingüística resultante do uso de conseqüentes não-lingüísticos.

A abordagem que está sendo proposta neste trabalho permite a escolha de um entre os dois modelos discutidos anteriormente. Entretanto, a ênfase de toda a etapa de aplicação em problemas práticos será nos modelos TS, e as razões para isto serão discutidas no capítulo 4, seção 4.4.

Capítulo 4

Abordagem Co-Evolutiva para o Projeto Automático de Sistemas Nebulosos

Sistemas nebulosos baseados em regras usam a teoria dos conjuntos nebulosos para descrever o conhecimento do especialista em vários domínios de aplicação (Pedrycz & Gomide, 1998). Entretanto, em muitos casos este conhecimento pode não estar facilmente disponível e a extração de conhecimento, a partir de bases de dados de dimensão elevada, nem sempre é tarefa fácil para um operador humano. Estas situações têm motivado o desenvolvimento de técnicas de extração automática e representação do conhecimento na forma de sistemas nebulosos, conhecidas como projeto automático de sistemas nebulosos.

No projeto de sistemas nebulosos, um aspecto fundamental é o balanço entre acuidade e interpretabilidade, levando-se em conta a autonomia de projeto. Algumas aplicações requerem sistemas nebulosos acurados. Para isso, o projetista deve considerar apenas o critério de erro. Contudo, nos últimos anos, a interpretabilidade do modelo tem recebido atenção especial. Quando a ênfase é a interpretabilidade, o projeto de sistemas nebulosos baseados em regras lingüísticas gera sistemas facilmente interpretáveis por um operador humano, embora o desempenho, em termos da acuidade na resposta, possa estar sujeito a alguma degradação. Ao lado da acuidade e interpretabilidade, um outro aspecto importante está relacionado à autonomia de projeto. Neste trabalho a autonomia será avaliada com base no grau de envolvimento do projetista ou usuário junto ao procedimento de modelagem adotado.

A solução de problemas práticos de alta complexidade, como é o caso do projeto automático de sistemas nebulosos, requer a construção de sistemas inteligentes, capazes de integrar coerentemente técnicas e metodologias originárias de diferentes áreas de pesquisa. As propostas de solução

para estes problemas devem considerar as vantagens de se utilizarem técnicas distintas de forma colaborativa, beneficiando-se do caráter complementar das metodologias disponíveis.

Recentemente, vários trabalhos e aplicações combinando a teoria dos conjuntos nebulosos e computação evolutiva têm sido propostos. Em particular, um grande número de trabalhos explora o uso de AGs no projeto de sistemas nebulosos. Estas técnicas híbridas conhecidas como *genetic fuzzy systems* (GFS) são, basicamente, sistemas nebulosos acrescidos de processos de aprendizagem baseados em AG (Cordón *et al.*, 2001b; Delgado *et al.*, 2001b; Herrera & Magdalena, 1998). GFS têm atraído considerável atenção de pesquisadores das mais diversas áreas, e isto se justifica pelo fato dos algoritmos genéticos corresponderem a ferramentas poderosas de otimização para o projeto de sistemas nebulosos, particularmente quando a maior parte das decisões de projeto podem ser codificadas em uma representação cromossômica.

Entretanto, quando decisões mais complexas devem ser tomadas, a evolução isolada de uma população de indivíduos, em que a solução completa do problema é codificada em um único indivíduo, pode se tornar inadequada. Nestes casos, pode ser mais apropriado co-evoluir os indivíduos das várias espécies.

A abordagem que será proposta neste capítulo é baseada em co-evolução e implementa diferentes espécies que codificam soluções parciais do problema de projeto de sistemas nebulosos. Relações de hierarquia e cooperação são estabelecidas entre indivíduos que representam diferentes parâmetros de um sistema nebuloso. Estes parâmetros são codificados em populações organizadas em quatro níveis distintos. Um esquema especial de avaliação de *fitness* é utilizado para se medir o desempenho de cada indivíduo. Restrições devem ser observadas em todos os níveis hierárquicos de modo a garantir a ocorrência de indivíduos válidos em termos da interpretabilidade do modelo obtido.

Este capítulo apresenta a abordagem co-evolutiva para o projeto automático de sistemas nebulosos e está dividido em quatro seções. A seção 4.1 destaca importantes aspectos do projeto de sistemas nebulosos. Na seção 4.2, algumas abordagens relacionadas ao projeto de sistemas nebulosos são discutidas, com destaque para aquelas baseadas em evolução, co-evolução e redes neurais, como o sistema neuro-nebuloso conhecido como ANFIS. A seção 4.3 introduz a proposta co-evolutiva, enfatizando as relações de hierarquia e cooperação, o esquema de codificação, o algoritmo e operadores evolutivos e, ainda, o esquema de avaliação do *fitness*. A seção 4.4 traz um exemplo do sistema nebuloso do tipo Takagi-Sugeno com consequentes não-lineares, descreve dois métodos para a otimização dos parâmetros dos consequentes e, por fim, propõe um algoritmo de poda para eliminação do excesso de flexibilidade ou redundância.

4.1 Projeto Automático de Sistemas Nebulosos

O projeto de sistemas nebulosos baseado em dados de entrada-saída tem atraído o interesse de diversos pesquisadores (Hoffmann & Nelles, 2000; Jang, 1993; Jin, 2000; Karr, 1991b; Nozaki *et al.*, 1997). Dentre importantes aspectos a serem observados, três características devem ser cuidadosamente analisadas: acuidade, interpretabilidade e a autonomia de projeto. A seguir, estas características serão discutidas:

- **acuidade:** O problema da acuidade pode ser crítico quando os modelos são usados, por exemplo, em sistemas de controle. Neste caso, o valor estimado é realimentado e pequenos erros podem se propagar, refletindo no comportamento ao longo do tempo (Espinosa & Vandewalle, 2000). A acuidade do modelo é usualmente medida por um critério de erro. Em problemas de aproximação de funções, a medida mais comum é dada pelo erro quadrático médio (EQM), definido como:

$$\text{EQM} = \frac{1}{N} \sum_{p=1}^N (y_p - \hat{y}_p)^2, \quad (4.2)$$

onde y_p é a saída desejada e \hat{y}_p é a saída estimada pelo sistema nebuloso, para o p -ésimo padrão de entrada, $p = 1, \dots, N$.

- **interpretabilidade:** um aspecto fundamental no projeto de sistemas nebulosos é a interpretabilidade ou transparência do modelo resultante. Isto significa que o conjunto de regras deve ser facilmente entendido por um operador humano (Jin, 2000). Entretanto, conforme ressaltado por Setnes *et al.* (1998), embora muitas abordagens na literatura salientem a importância da interpretabilidade do modelo, na realidade, usam os sistemas nebulosos como verdadeiras caixas-pretas, para as quais a acuidade é o único objetivo e nenhum critério de interpretabilidade é considerado. Apesar do grande apelo da interpretabilidade em modelos nebulosos, não existe nenhuma definição bem estabelecida para este conceito (Jin, 2000). Aqui, a interpretabilidade será identificada e analisada com base em quatro características distintas:

- **visibilidade:** O projeto de sistemas nebulosos pode ser utilizado para obter não só modelos acurados, mas também bases de regras cujas partições nebulosas apresentem um alto grau de visibilidade. Na abordagem que está sendo proposta, a visibilidade está associada a dois parâmetros distintos (γ e κ), que definem as propriedades de γ -completude e κ -sobreposição definidas a seguir:

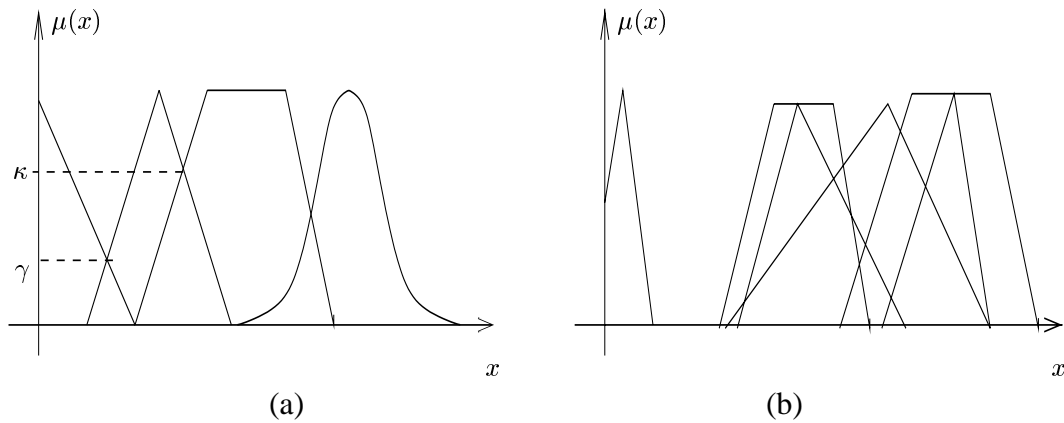


Figura 4.1: Análise de interpretabilidade em partições não-uniformes: (a) partição nebulosa interpretável; (b) partição nebulosa não interpretável.

- * γ -completude. Para esta propriedade, também conhecida como propriedade de cobertura do universo (Espinosa & Vandewalle, 2000), fixa-se um grau mínimo (γ) de sobreposição entre as funções de pertinência na partição do universo, de modo a garantir partições sem falhas (Gonzales & Perez, 1998; Jang, 1993). Esta propriedade garante que a base de regras cobrirá todas as regiões do universo das variáveis envolvidas.
- * κ -sobreposição. Esta propriedade estabelece um grau máximo (κ) de sobreposição na partição do universo (Delgado *et al.*, 2001a), evitando-se assim partições com funções de pertinência completamente sobrepostas.

Estas condições garantem que, dado um valor x qualquer no intervalo de operação de uma das variáveis de entrada, é possível se encontrar uma função de pertinência A tal que $\mu_A(x) \geq \gamma$, e no máximo uma função de pertinência B , tal que $\mu_B(x) > \kappa$. A figura 4.1 traz uma análise da interpretabilidade, do ponto de vista de visibilidade, para dois exemplos arbitrários de partições nebulosas. Para partições uniformes do universo, tem-se $\gamma = \kappa$.

- **simplicidade:** Regras lingüísticas longas, i.e., regras com muitas condições no antecedente, não são facilmente compreendidas (Ishibuchi *et al.*, 2001). Conforme discutido por Delgado *et al.* (2001a), a inclusão de condições irrelevantes (*don't care conditions*) (Ishibuchi & Nakashima, 1999), além de simplificar as regras nebulosas, melhora a capacidade de generalização do sistema resultante. A simplicidade de cada regra é avaliada por seu comprimento, medido pelo número de variáveis presentes na regra,

decrecido do número de variáveis irrelevantes.

- **compactação:** O tamanho da base de regras é inversamente proporcional à interpretabilidade do modelo. O grau de compactação da base de regras é medido pelo número efetivo de regras que compõem o sistema nebuloso (Ishibuchi *et al.*, 2001).
- **consistência:** A condição de consistência, definida no contexto de classificação (Gonzales & Perez, 1998), é aqui adaptada para eliminar as regras conflitantes na base de regras. Então, a condição de consistência adotada determina a exclusão de regras com o mesmo antecedente e consequentes diferentes.
- **autonomia de projeto:** Sistemas automáticos são sistemas com dispositivos de auto-regulamentação, formados por leis e estratégias com as quais controlam seu comportamento. Para ser autônomo, um sistema precisa primeiramente ser automático, mas a autonomia depende ainda da capacidade de desenvolver suas próprias estratégias de controle (Steels, 1995). Na abordagem a ser proposta, a autonomia no projeto será avaliada pelo grau com que a metodologia para o desenvolvimento dos sistemas nebulosos independe da intervenção do usuário ou projetista. Quando muitos parâmetros do modelo devem ser estabelecidos pelo usuário, forçando-o inclusive a arbitrar certos parâmetros em virtude de não haver informação suficiente que permita deduzir seus valores ótimos, o projeto do sistema nebuloso é considerado não-automático, e portanto, não-autônomo. Além disso, quando o sistema tem tempo e condições para analisar um grande número de exemplos, ou especular sobre como poderia lidar com novas situações, como é o caso de sistemas baseados em treinamento, a total autonomia do projeto pode ficar comprometida. Neste sentido, abordagens para projetar sistemas nebulosos com base em padrões de entrada-saída, para as quais a intervenção do usuário é mínima e nenhuma informação *a priori* sobre a estrutura final do sistema é fornecida, podem ser consideradas metodologias semi-autônomas.

4.2 Abordagens Relacionadas

Muitas abordagens têm sido propostas para o projeto de sistemas nebulosos. Entretanto, não existe uma metodologia definitiva que maximize, ao mesmo tempo, a acuidade e interpretabilidade do modelo, mantendo a completa autonomia de projeto.

Um dos primeiros trabalhos relevantes na linha de projeto de sistemas nebulosos foi proposto por Nomura *et al.* (1991). Neste artigo, os autores utilizam um mecanismo de auto-ajuste de regras

nebulosas, baseado no método do gradiente. O método otimiza, simultaneamente, o antecedente das regras e os parâmetros dos conseqüentes de sistemas nebulosos do tipo Takagi-Sugeno (TS). Funções de pertinência triangulares e conseqüentes constantes são ajustados de forma automática, e o produto algébrico é fixado como o operador de agregação dos antecedentes (A_a definido no capítulo 3, seção 3.2.2) das regras nebulosas.

Muitas abordagens neuro-nebulosas propostas na literatura (Jang, 1993; Lin & Lee, 1996) também utilizam o método do gradiente para otimizar alguns parâmetros dos sistemas nebulosos. Entretanto, na maioria dos casos, parâmetros críticos do projeto devem ser fornecidos pelo usuário. E mais, o uso de técnicas de otimização sem nenhuma restrição associada à configuração final do sistema pode, em alguns casos, resultar em partições nebulosas com funções de pertinência completamente sobrepostas ou regiões com ausência de sobreposição. Isto acaba por comprometer a consistência e a interpretabilidade da base de regras, dificultando a compreensão do modelo.

Uma proposta neuro-nebulosa bastante difundida na literatura é o ANFIS - *Adaptative Neuro Fuzzy Inference System* (Jang, 1993), que utiliza um algoritmo de aprendizagem híbrido para estimar os parâmetros de sistemas nebulosos TS. Jang (1993) propõe um modelo nebuloso organizado numa arquitetura do tipo rede neural, no qual técnicas de otimização são usadas para ajustar os parâmetros do antecedente e do conseqüente. O modelo ANFIS ajusta, através do algoritmo *back-propagation*, os parâmetros (formato e localização) das funções de pertinência, as quais serão de um tipo único (todas triangulares, todas Gaussianas, etc), dependendo da escolha do usuário. O método dos quadrados mínimos recursivo é usado para se encontrarem os coeficientes das funções lineares, que formam o conseqüente das regras nebulosas.

O ANFIS é considerado uma referência na área de projeto de sistemas nebulosos. Entretanto, este sistema apresenta algumas desvantagens. Como apontado por Rizzi *et al.* (1999), o uso do ANFIS é restringido pela quantidade de parâmetros a serem definidos como: o tipo das funções de pertinência, a granularidade dos universos, que, por sua vez, fixa o total de regras nebulosas e, por fim, o operador de agregação dos antecedentes (A_a). Todos estes parâmetros dependem da intervenção do usuário e nenhuma informação é fornecida indicando qual a melhor escolha para determinada aplicação. Como alternativa, Rizzi *et al.* (1999) propõem um modelo de aprendizado baseado em agrupamento (*clustering*) para se extrair um conjunto de regras nebulosas de modelos TS. A proposta utiliza a mesma estrutura do ANFIS, mas o aprendizado é automático, eliminando-se assim a necessidade de intervenção do usuário na definição de parâmetros críticos. Uma outra desvantagem do ANFIS está relacionada à interpretabilidade do modelo. No processo de ajuste adotado pelo ANFIS, por exemplo, a otimização irrestrita pode dificultar a obtenção de partições

nebulosas que atendam o critério de γ -*completude*, e κ -*sobreposição*, para qualquer $\gamma > 0$ e $\kappa < 1$. Por fim, conforme discutido por Chen & Liou (1999), o problema dos mínimos locais, resultante do uso do método de otimização baseado no gradiente, também adotado pelo ANFIS, pode prejudicar o aprendizado. Chen & Liou (1999) apresentam uma proposta para aumentar a velocidade de convergência do ANFIS, mas o grau de intervenção do usuário permanece o mesmo do modelo original proposto por Jang (1993).

Inúmeras abordagens apresentadas na literatura empregam algoritmos genéticos para a obtenção de diversos parâmetros de sistemas nebulosos e são conhecidas como *Genetic Fuzzy System* (GFS). Dentre as primeiras propostas na linha de GFS, lançadas no início dos anos 90, destacam-se os trabalhos de Karr (1991b) e Thrift (1991). Uma parte das propostas de GFS ajusta apenas um subconjunto dos parâmetros, evoluindo somente parâmetros das funções de pertinência ou apenas a estrutura das regras. Por exemplo, Thrift (1991) utilizou um AG para selecionar conjuntos nebulosos na parte do conseqüente das regras nebulosas. O formato das funções de pertinência é fixo e a evolução define apenas quais regras farão parte do modelo.

Outras propostas baseadas em AG tentam ajustar tanto os parâmetros do antecedente quanto as regras nebulosas, resultando em abordagens com maior grau de autonomia (Delgado *et al.*, 2001a; Homaifar & McCormick, 1995; Karr, 1991a; Siarry & Guely, 1998). Em Karr (1991a), funções de pertinência e regras são determinadas por evolução através de um AG com codificação binária. Primeiramente, a mutação e o crossover são aplicados para se determinar o número de regras nebulosas, com o formato de cada função de pertinência mantido fixo. Somente num segundo estágio os parâmetros das funções são ajustados pelo AG. Em uma proposta alternativa, Homaifar & McCormick (1995) usaram cromossomos codificados por valores inteiros. Após a decodificação, estes cromossomos definem valores lingüísticos das variáveis de entrada e funções de pertinência associadas. Para as variáveis de saída, as funções de pertinência são fixas. O grau de sobreposição entre os conjuntos é variável, mas um limite mínimo é obedecido. Como nenhum limite máximo é imposto, partições com funções completamente sobrepostas podem emergir do processo evolutivo. Em outro exemplo de GFS, Siarry & Guely (1998) utilizaram um AG para ajustar os parâmetros dos sistemas nebulosos em que todas as funções de pertinência são do tipo triangular (com parâmetros ajustáveis) e os conseqüentes TS são constantes. Com o objetivo de aumentar a autonomia de projeto, Delgado *et al.* (2001a) propuseram o uso de modularidade e hierarquia associada a algoritmos genéticos para ajustar uma série de parâmetros de modelos TS, conhecida como sistemas nebulosos genéticos hierárquicos (HGFS). Na abordagem HGFS, o método de otimização global (a ser descrito na seção 4.4.2) é utilizado para o ajuste dos conseqüentes.

Na maioria das propostas citadas anteriormente, a acuidade é o principal objetivo no projeto dos sistemas nebulosos. Entretanto, em muitas abordagens apresentadas atualmente, a interpretabilidade do modelo tem recebido atenção especial (Espinosa & Vandewalle, 2000; Ishibuchi & Nakashima, 1999; Jin, 2000; Setnes *et al.*, 1998; Yen *et al.*, 1998). Uma discussão bem interessante sobre o compromisso entre acuidade e interpretabilidade é apresentada em (Espinosa & Vandewalle, 2000). Espinosa & Vandewalle (2000) apresentaram um paradigma para extrair regras nebulosas com integridade lingüística. O algoritmo é baseado numa abordagem de duas etapas: num primeiro passo, utilizam-se técnicas de agrupamento e projeção para se encontrarem boas posições iniciais para os conjuntos nebulosos nos domínios de entrada; num segundo passo, a complexidade do modelo é reduzida utilizando-se o conceito de integridade semântica. Valores reais na parte do conseqüente, obtidos pelo método dos quadrados mínimos, são posteriormente transformados em termos lingüísticos. Ishibuchi & Nakashima (1999) apresentaram uma abordagem baseada em AG para extrair um conjunto reduzido de regras nebulosas do modelo Mamdani, com claro significado lingüístico. Em outro exemplo cujo enfoque é a interpretabilidade, Jin (2000) propôs uma abordagem eficiente para o projeto de sistemas nebulosos em espaços de dimensões elevadas. A estrutura e os parâmetros dos sistemas nebulosos são otimizados utilizando-se AG e um método baseado no gradiente. O sistema nebuloso resultante é interpretável, relativamente simples e as dependências entre as variáveis de entrada e saída são claramente descritas por funções de conseqüentes constantes. Setnes *et al.* (1998) discutiram a importância do compromisso entre acuidade e transparência para que um sistema nebuloso se distancie de uma caixa-preta. Nesta abordagem, um processo de agrupamento no espaço de entradas gera a estrutura das regras nebulosas. Os parâmetros dos conseqüentes TS, expressos por *singletons*, são transformados em termos lingüísticos, aumentando a compreensão do modelo.

Como um exemplo de modularidade baseada na idéia de soluções parciais evoluindo em diferentes populações, Heider & Drabe (1997) propuseram um AG em cascata, cuja execução é dividida em dois laços: um laço (ou cascata) interno e outro externo que, juntos, determinam e ajustam os parâmetros e a estrutura dos sistemas nebulosos. Os laços utilizam AG para evoluir as funções de pertinência em um nível e a base de regras no outro. Para Heider & Drabe (1997), a implementação de duas populações diferentes, codificadas em dois níveis distintos, produz considerável melhora no desempenho dos sistemas nebulosos, quando comparados com sistemas produzidos por técnicas convencionais de AGs.

O uso de co-evolução para a solução de problemas complexos tem surgido como alternativa às técnicas evolutivas tradicionais. Neste sentido, dois importantes trabalhos a serem citados são

as abordagens propostas por Moriarty & Miikkulainen (1997, 1998) e Potter & De Jong (2000), apresentadas na seção 2.4. Moriarty & Miikkulainen (1997) empregaram uma evolução simbiótica para a construção de redes neurais artificiais e Moriarty & Miikkulainen (1998) utilizaram a hierarquia para a obtenção dos modelos das redes neurais. O modelo cooperativo proposto por Potter & De Jong (2000) utiliza a co-evolução para solucionar diferentes problemas, incluindo o conhecido problema de classificação das espirais concêntricas, que será tratado no capítulo 5.

A tabela 4.1 resume as características de algumas das abordagens para o projeto automático de sistemas nebulosos discutidas anteriormente. A tabela apresenta as características de cada abordagem da seguinte forma:

- **Método:** especifica a técnica (evolução (evol), gradiente (grad), outros) que a abordagem utiliza para obter os sistemas nebulosos, mostrando ainda qual o algoritmo evolutivo utilizado (algoritmos genéticos (AG), programação genética (PG), etc);
- **Modelo:** especifica o modelo de sistema nebuloso obtido pela abordagem: Mamdani ou TS;
- **Codificação (Cod):** especifica o tipo de codificação utilizada nos processos evolutivos: binária, inteira, real ou mista;
- **Enfoque:** mostra o enfoque do trabalho em relação à acuidade (acuid) e interpretabilidade (interp) do modelo obtido;
- **Agregação dos Antecedentes (A_a):** especifica a forma de obtenção do operador de agregação dos antecedentes: usuário (user), automática (autom) ou fixada pela abordagem (fixa);
- **Base de Regras (BR):** especifica a forma como é obtida a base de regras do sistema nebuloso: automática (autom) ou fixada pela abordagem (fixa), como é o caso das abordagens que trabalham com todas as regras, sem a possibilidade de exclusão;
- **Funções de Pertinência (FP):** especifica a forma de obtenção das funções de pertinência: automática (autom) ou fixada pela abordagem (fixa); e o formato que as funções podem assumir: triangular (trg), trapezoidal (trp) ou Gaussiana (gauss);
- **Conseqüentes TS (Conseq):** especifica o tipo de função utilizada no conseqüente das regras para os modelos TS: não-linear (nl), linear (lin) ou constante (ct);
- **Método de Otimização do Conseqüente TS (Otim):** o método utilizado para se estimarem os parâmetros da função no conseqüente TS: global, local, gradiente (grad), evolução (evol) ou por uma heurística própria (heur);

4.3 Sistema Nebuloso Co-Evolutivo: Uma Proposta Baseada em Hierarquia e Cooperação

A abordagem proposta neste capítulo se baseia no esquema de evolução hierárquica sugerido por Delgado *et al.* (2001a). A inspiração do processo co-evolutivo adotada aqui vem do paradigma baseado em hierarquia idealizado por Moriarty & Miikkulainen (1998) e do modelo de co-evolução cooperativa apresentado por Potter & De Jong (2000). Populações de diferentes níveis hierárquicos são interpretadas como espécies distintas ou módulos distintos, que codificam soluções parciais do problema de projeto de sistemas nebulosos. A estrutura hierárquica pode ser implementada de modo que indivíduos em diferentes módulos mantenham uma relação cooperativa e o *fitness* de um indivíduo dependa do *fitness* de indivíduos pertencentes a outros módulos. Por meio destas relações cooperativas envolvendo indivíduos de diferentes espécies, os parâmetros dos sistemas nebulosos são evoluídos.

4.3.1 A Abordagem Co-evolutiva

A abordagem co-evolutiva supõe populações distintas, organizadas de forma hierárquica em módulos evolutivos. Indivíduos de diferentes populações, representando quatro espécies distintas, codificam diferentes parâmetros dos sistemas nebulosos. O esquema básico da co-evolução é mostrado na figura 4.2. Em princípio, qualquer algoritmo evolutivo apropriado pode ser usado, mas aqui os algoritmos genéticos serão enfatizados.

A estrutura modular, hierárquica e cooperativa que está sendo proposta considera:

- uma população de partições nebulosas no primeiro nível;
- uma população de regras individuais no segundo nível;
- uma população de bases de regras no terceiro nível;
- uma população de sistemas nebulosos no quarto e último nível.

Indivíduos dos níveis superiores são construídos a partir de indivíduos de níveis inferiores. Isto implica que indivíduos de um nível cooperam por meio de relações de interdependência na instanciação de indivíduos de outros níveis, como será detalhado na seção a seguir.

Tabela 4.1: Comparação das abordagens para o projeto automático de sistemas nebulosos

Abordagem	Método	Modelo	Codificação	Enfoque	A_a	BR	FP	Conseq	Otim
Delgado <i>et al.</i> (2000a)	evol(AG)	TS	mista	acuid	autom	autom	autom	nl	global
Espinosa & Vandewalle (2000)	outros	TS	-	interp	user	autom	autom(trg)	ct	global
Heider & Drabe (1997)	evol modular(AG)	Mamdani	binária	acuid	fixa	autom	autom(trp)	-	-
Hoffmann & Nelles (2000)	evol(PG)	TS	mista(PG)	acuid	user	autom	autom(gauss)	lin	local
Homaifar & McCormick (1995)	evol(AG)	Mamdani	inteira	acuid	fixa	autom	autom(trg)	-	-
Ishibuchi & Nakashima (1999)	evol(AG)	Mamdani	binária	interp	fixa(*) ^a	autom	fixa(trg)	-	-
Jang (1993)	grad	TS	-	acuid	user	fixa	autom	lin	global
Jin (2000)	outros	TS	-	interp	fixa(*)	autom	autom(gauss)	ct	grad
Karr (1991a)	evol(AG)	Mamdani	binária	acuid	fixa	autom	autom	-	-
Nomura <i>et al.</i> (1991)	outros	TS	-	acuid	fixa(*)	fixa	autom(trg)	ct	heur
Setnes <i>et al.</i> (1998)	outros	TS	-	interp	fixa(*)	autom	autom(gauss)	lin	global
Shi <i>et al.</i> (1999)	grad	TS	mista	acuid	fixa(*)	fixa	autom	ct	grad
Siarry & Gueley (1998)	evol(AG)	TS	binária	acuid	fixa(*)	autom	autom(trg)	ct	evol
Thrift (1991)	evol(AG)	Mamdani	inteira	acuid	user	autom	fixa(trg)	-	-

^aproduto algébrico

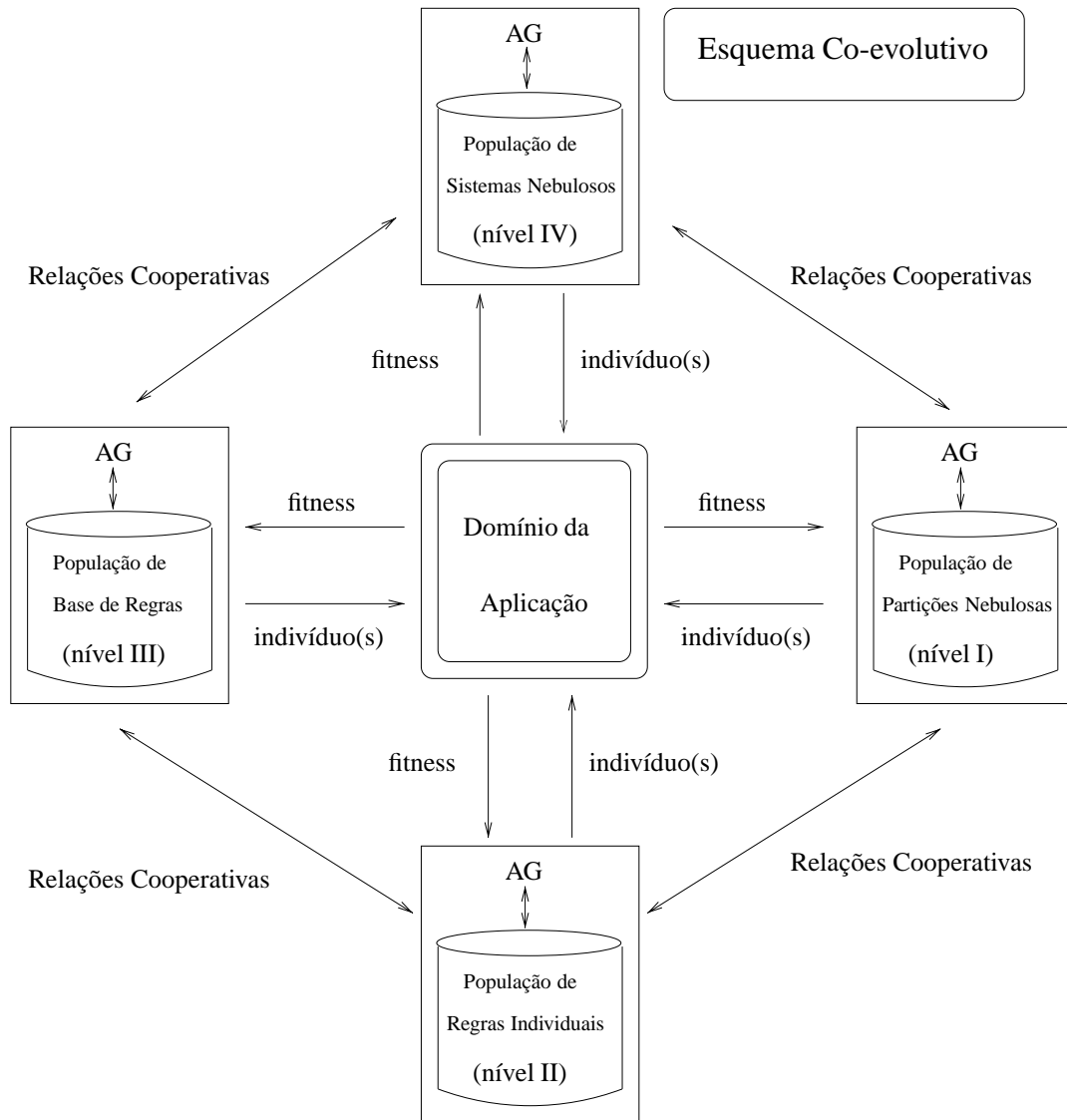


Figura 4.2: O modelo co-evolutivo para evoluir os sistemas nebulosos.

4.3.2 Hierarquia e Cooperação

Um esquema de cooperação emerge das relações de interdependência entre indivíduos de diferentes espécies ou módulos evolutivos. Os indivíduos das partições nebulosas (nível I) representam as funções de pertinência nas granularizações dos universos das variáveis envolvidas. Estes indivíduos são codificados em cromossomos com alelos inteiros e reais (codificação mista). Cada membro da população de regras individuais (nível II) representa uma regra nebulosa codificada por cromossomos com alelos que assumem valores inteiros. Esta população aceita diferentes combinações das funções de pertinência identificadas por seus índices, i.e., a ordem na partição nebulosa. Valores nulos são possíveis e codificam a irrelevância da variável na respectiva regra. Neste caso, estes indivíduos representam regras genéricas (ver capítulo 3, seção 3.2.2). A população de base de regras (nível III) tem indivíduos formados por índices que identificam as regras individuais correspondentes. O comprimento do cromossomo é fixo e determina o número máximo de regras nebulosas. Cada indivíduo do nível IV (codificação mista) representa um sistema nebuloso. A figura 4.3 ilustra o cromossomo representando um indivíduo na população.

Alelo no Sistema Nebuloso (nível IV)

1	→	Inferência (Escalonada)
2	→	Agregação dos Antecedentes (t-norma)
3	→	parâmetro p_t
4	→	Semântica da Regra (Norma)
5	→	parâmetro associado
6	→	Agregação das Regras
7	→	<i>Defuzzification</i>
8	→	Índice da Base de Regras
9	→	Índice da Partição Nebulosa

Figura 4.3: Esquema de codificação dos sistemas nebulosos no nível IV.

Neste nível, o código de cada cromossomo associa uma base de regras (alelo no *locus* 8) e uma partição nebulosa (alelo no *locus* 9), a um conjunto de operadores para definir a semântica da regra e operações específicas do processo de inferência (alelos nos *loci* 1 a 7). Inicialmente, trabalhou-se

com uma única partição nebulosa no nível I tornando desnecessário o alelo no *locus* 9. Entretanto, testes realizados com uma população de partições nebulosas trouxeram melhores resultados e a informação sobre qual das partições deveria ser utilizada passou a ser necessária. Para modelos TS, os alelos localizados nos *loci* 4 a 7, cujo propósito é o tratamento de outras classes de modelos de sistemas nebulosos, não são considerados. Para os modelos Mamdani, por exemplo, os alelos nos *loci* 4 e 5 identificam os operadores de semântica das regras (norma e parâmetro associado), o alelo no *locus* 6 representa o operador de agregação das regras A_R (definido no capítulo 3, seção 3.2.2), e o alelo no *locus* 7 define o método de transformação da saída nebulosa em não-nebulosa (definido no capítulo 3, seção 3.3.1). Para todos os modelos, a agregação do antecedente (A_a) é dada pelos alelos nos *loci* 2 e 3 (norma-t e parâmetro p_t associado).

A figura 4.4(a) traz uma visão geral destas relações e a figura 4.4(b) mostra os detalhes da hierarquia e cooperação que guiam o processo co-evolutivo. No exemplo da figura 4.4, o i -ésimo sistema nebuloso mostrado no nível IV utiliza a 30ª base de regras do nível III e o 4º indivíduo da população de partições nebulosas do nível I.

Portanto, na composição do i -ésimo sistema nebuloso, cada regra individual da 30ª base de regras agrega as proposições de entrada nos antecedentes das regras nebulosas através da t-norma t_5 , apresentada na seção 3.1.3. Neste caso particular, a 15ª regra na população de regras individuais seria dada por:

Se (X_1 é “2”) t_5 (X_2 é “1”) então Y é $g(\cdot)$, para modelos TS;

e

Se (X_1 é “2”) t_5 (X_2 é “1”) então Y é “5”, para modelos Mamdani,

onde $g(\cdot)$ representa uma função não-linear a ser detalhada na seção 4.4 e os termos lingüísticos indexados por “2” e “1” e eventualmente “5” para os modelos Mamdani são, neste caso, definidos no 4º indivíduo do nível I. Os detalhes da codificação no nível I serão apresentados na seção a seguir.

As relações de interdependência, mostradas na figura 4.4 e discutidas anteriormente, imprimem à abordagem proposta uma complexidade adicional, que não permite classificá-la nem como *Pittsburgh* nem como *Michigan* (ver capítulo 2, seção 2.2.2). Neste sentido, a abordagem pode ser classificada como uma abordagem mista entre as abordagens *Pittsburgh* e *Michigan*, já que nos níveis I, II e IV, um único indivíduo de cada população é utilizado na composição da solução e no nível III, um conjunto de indivíduos compõe a solução do problema.

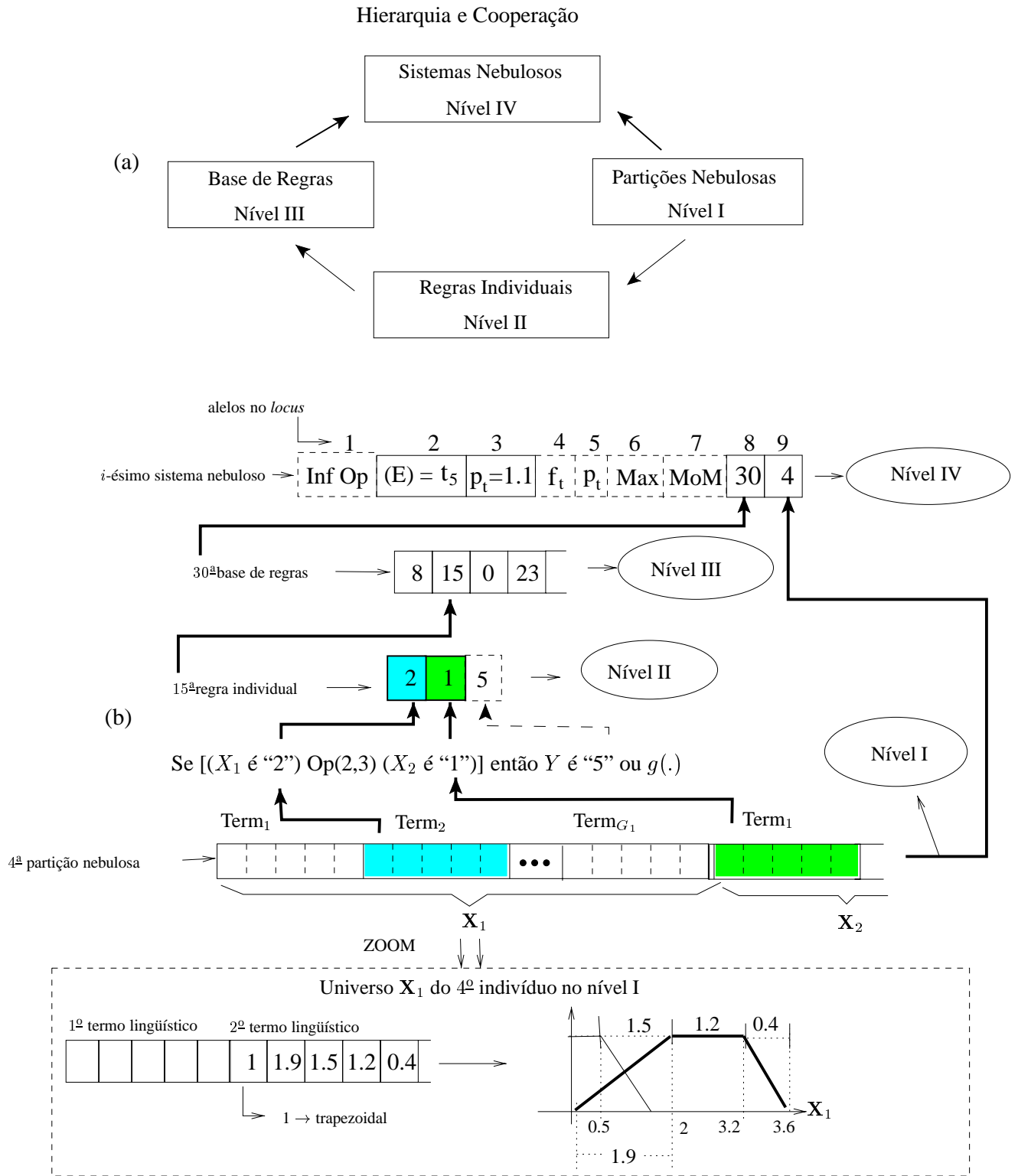


Figura 4.4: Hierarquia e cooperação entre: (a) espécies; (b) indivíduos de espécies diferentes.

Esquema de Codificação das Funções de Pertinência

O esquema de codificação adotado no nível I (partições nebulosas) se baseia no conceito de posições relativas e é análogo ao esquema de codificação de funções de pertinência triangulares descritos em Lee & Takagi (1993). Uma vez que o cromossomo é definido, um processo de decodificação é necessário para se gerar o fenótipo da função de pertinência (Delgado *et al.*, 2001a).

A escolha deste tipo de codificação se baseia em dois argumentos principais:

- uniformidade: três tipos de funções de pertinência são possíveis (trapezoidal, triangular e Gaussianas) e cada tipo necessita de um total de parâmetros diferentes (conforme visto no capítulo 3, seção 3.1.1). Portanto, optou-se por um esquema uniforme de codificação baseado em 5 alelos que, associado a um processo de decodificação, possibilitou a implementação destes três tipos.
- relatividade: com o objetivo de reduzir o total de indivíduos inválidos resultantes da operação de crossover, optou-se pelo esquema de codificação relativo que será descrito a seguir.

A figura 4.5 traz os detalhes da relação entre genótipo e fenótipo, que ocorre no nível I.

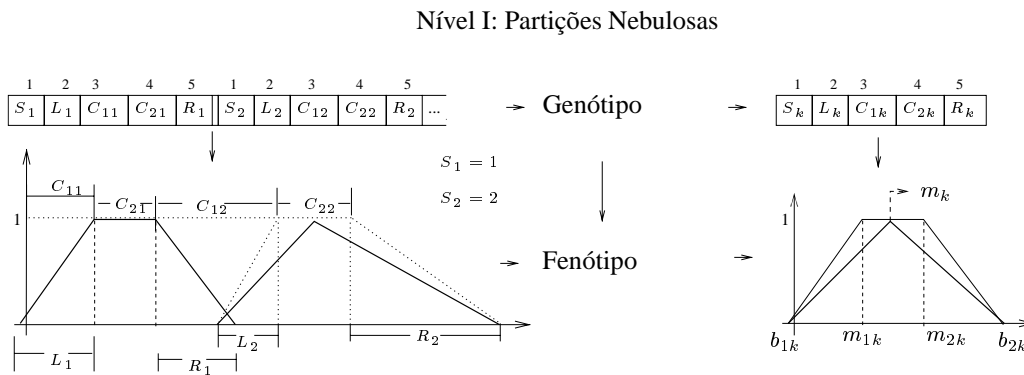


Figura 4.5: Esquema de Codificação das Funções de Pertinência Trapezoidal e Triangular.

As funções de pertinência (FP) são representadas por uma seqüência de cinco alelos

$$\{S_k, L_k, C_{1k}, C_{2k}, R_k\},$$

onde k identifica a posição da função no universo associado. As FP podem assumir três tipos diferentes: (1) trapezoidal, (2) triangular, e (3) Gaussiana. O tipo da função é definido pelo alelo

(S_k), localizado no *locus* 1 da seqüência de cinco alelos. Valores nulos no primeiro *locus* ($S_k = 0$) identificam as funções de irrelevância (definidas no capítulo 3, seção 3.1.1). Com exceção do alelo que representa o tipo da função (que assume valores inteiros), cada alelo da seqüência de 5 assume valores reais que são relativos a uma posição de referência que ocorre no *locus* 3 (C_{1k}) e mede a distância da função de pertinência k para a função de pertinência $k - 1$ (no caso de $k = 1$ a distância é medida em relação ao limite esquerdo do universo). Logo, a posição absoluta de um termo específico no universo depende da posição do termo imediatamente anterior. Para a k -ésima função de pertinência, os seus valores absolutos seriam dados por: $m_{1k} = m_{2k-1} + C_{1k}$; $m_{2k} = m_{1k} + C_{2k}$; $b_{1k} = m_{1k} - L_k$; $b_{2k} = m_{2k} + R_k$. Para a obtenção do fenótipo das funções do tipo trapezoidal, os alelos nos *loci* 2 a 5 são convertidos em seus valores absolutos $\{b_{1k}, m_{1k}, m_{2k}, b_{2k}\}$ que definem os parâmetros $\{a, m, n, b\}$ destas funções (ver capítulo 3, seção 3.1.1). No caso de funções triangulares, após a determinação dos valores absolutos, o centro m_k é encontrado pela média entre m_{1k} e m_{2k} . Para funções Gaussianas, o valor modal é calculado da mesma forma que o centro das funções triangulares, mas uma modificação é necessária para o cálculo da dispersão: a dispersão é definida como $\sigma_k = \text{Disp}_k^{\max}/3$, onde o valor de Disp_k^{\max} , que identifica a dispersão máxima, é dado por $\text{Disp}_k^{\max} = (L_k + R_k)/2$. Portanto, os limites práticos das funções de pertinência do tipo Gaussiana não excederão $3\sigma_k$. A figura 4.6 ilustra o esquema de codificação adotado.

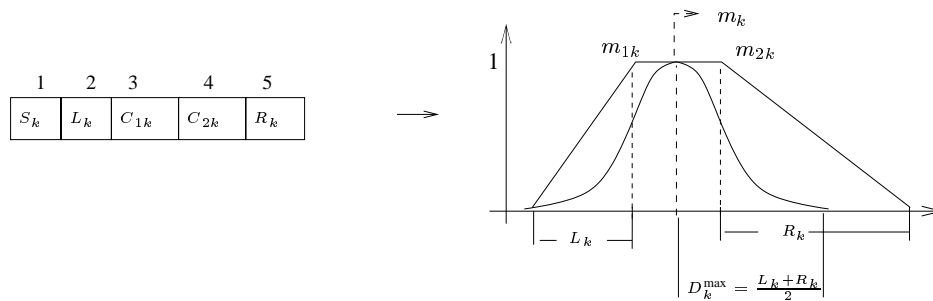


Figura 4.6: Esquema de Codificação da Função de Pertinência Gaussiana.

4.3.3 Algoritmo Genético Proposto

As espécies co-evoluem através de aplicações repetidas dos operadores evolutivos, conforme ilustrado pelo diagrama da figura 4.7.

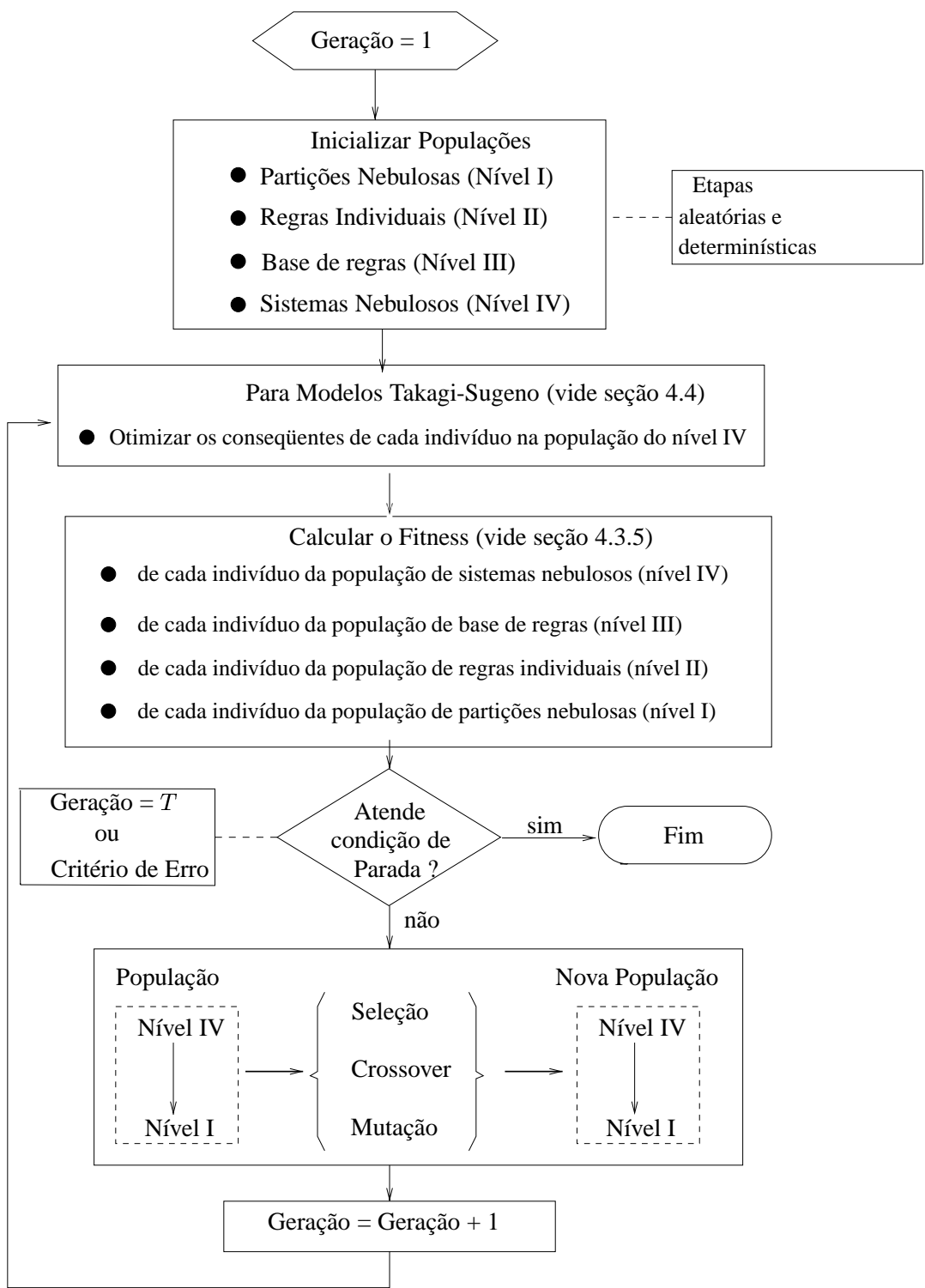


Figura 4.7: Algoritmo genético utilizado no processo co-evolutivo.

A etapa de inicialização compreende a geração de populações determinísticas ou aleatórias, dependendo do nível:

- Nível I: as partições nebulosas (PN) são representadas por cromossomos com codificação real que são inicializados de forma a gerar partições uniformes dos universos associados. Esta condição de uniformidade é alcançada atribuindo-se os mesmos valores para os alelos S_k , L_k , R_k e C_{2k} e escolhendo os valores de C_{1k} mais adequados para distribuir de maneira equidistante, todos os termos definidos pela granularidade máxima G_l em cada universo associado à variável x_l .
- Nível II: as regras nebulosas individuais (RI) são representadas por cromossomos com codificação inteira, que são gerados de forma aleatória. No caso da possibilidade de regras genéricas, as condições irrelevantes, codificadas por valores nulos, são introduzidas para produzir regras com comprimento reduzido, i.e., regras simplificadas (com menos termos lingüísticos no antecedente). A única restrição imposta é a de que todos os termos lingüísticos associados a cada variável devem estar presentes na população de regras individuais.
- Nível III: as bases de regras (BR) são representadas por cromossomos de tamanho fixo, com codificação inteira e são inicializados de forma a garantir a presença de todas as funções de pertinência na respectiva BR. Esta condição é alcançada através da análise de quais regras individuais devem ser incluídas para completar os índices dos termos lingüísticos que estão faltando. Exclusões de regras ocorrem atribuindo-se valores nulos aos alelos correspondentes.
- Nível IV: os sistemas nebulosos (SN) são representados por cromossomo com codificação mista e são inicializados com valores aleatórios, respeitando o tipo de valores que cada alelo pode assumir (a única exceção à geração aleatória é o parâmetros p_t). No caso de sistemas nebulosos do tipo TS, somente os alelos nos *loci* 2,3,8 e 9 são considerados. Alelos inteiros no *locus* 2 são gerados, de modo a cobrir todas as possíveis t-normas descritas na seção 3.1.3, que representam os operadores de agregação do antecedente das regras nebulosas A_a . Alelos no *locus* 3 são inicializados com o valor $p_t = 2.0$. Este valor foi escolhido arbitrariamente, mas testes com outros valores indicaram que a inicialização deste parâmetro afeta pouco o resultado final. Alelos nos *loci* 8 e 9 são inicializados aleatoriamente e definem que base de regras e partição nebulosa serão utilizada no respectivo sistema nebuloso.

As restrições impostas pela geração determinística na etapa de inicialização garantem, além

de partições uniformes nos universos das variáveis envolvidas, a presença de cada um dos termos lingüísticos possíveis em pelo menos uma das regras que compõem os sistemas nebulosos no nível IV.

Após a inicialização das populações em todos os níveis, o próximo passo envolve a otimização dos parâmetros dos conseqüentes, em cada um dos sistemas nebulosos do nível IV. Este procedimento, que será detalhado na seção 4.4, só é executado no caso de modelos TS. Quando o processo evolutivo utiliza os modelos Mamdani, este passo é desconsiderado, pois os conseqüentes são representados por conjuntos nebulosos codificados no nível I e ajustados pelos operadores evolutivos.

O passo seguinte, que compreende o cálculo do fitness, tem início com a avaliação de desempenho de cada sistema nebuloso, no domínio da aplicação a ser considerada. Os detalhes do cálculo de fitness serão apresentados na seção 4.3.5.

No último passo, a condição de parada (número máximo de gerações ou critério de erro) é testada. Caso esta condição não seja atendida, aplicam-se os operadores evolutivos. A seqüência de aplicação dos operadores evolutivos é do topo para a base, ou seja, seleção, crossover e mutação são primeiramente aplicados a todos os indivíduos do nível IV, para então serem aplicados aos indivíduos dos níveis hierarquicamente inferiores. Outras seqüências foram testadas (base para o topo, escolha aleatória a cada geração, etc.) mas os melhores resultados foram obtidos com a seqüência do topo para a base, embora não se tenha detectado uma razão específica para justificar este fato.

4.3.4 Operadores Evolutivos

A seleção é o primeiro operador evolutivo empregado e utiliza a técnica do torneio para selecionar 80% dos indivíduos. Nesta técnica, escolhe-se uma porcentagem de cromossomos (10%), selecionados aleatoriamente para participar em cada torneio e o melhor indivíduo (aquele com o maior valor de *fitness*) vence. O restante dos indivíduos da população (20%) é escolhido pelo método de seleção por diversidade. Aqui, o critério de diversidade focaliza a escolha nos cromossomos “mais diferentes”, quando comparados com o melhor indivíduo. A medida de diversidade é dada pela distância Euclidiana, para cromossomos com valores reais, e distância de Hamming, para cromossomos com codificação inteira. O processo de seleção é combinado com o elitismo, no qual o cromossomo com maior *fitness* sempre sobrevive para compor a próxima geração.

O segundo operador evolutivo é o operador genético de crossover. Os pares de crossover são escolhidos com base no *fitness* (os melhores indivíduos têm maior chance de serem escolhidos).

Em geral, aplica-se o crossover simples, com o ponto de crossover escolhido de forma aleatória, mas outros operadores de crossover podem ser usados. O crossover uniforme também foi testado mas, na maioria dos casos, o desempenho com crossover simples foi superior.

A mutação, o último operador evolutivo, é aplicada aos cromossomos com codificação real, inteira ou mista. Para cromossomos com codificação inteira, o novo valor é escolhido dentro de um conjunto de valores possíveis. Para a codificação em valores reais, a mutação uniforme ou não uniforme (ver seção 2.3.3) é aplicada. As seções anteriores evidenciaram o fato de que os cromossomos, em alguns níveis, contêm alelos que assumem valores inteiros e outros alelos que assumem valores reais. Obviamente, o tipo de mutação vai depender do tipo de valor assumido pelo alelo escolhido. A abordagem co-evolutiva proposta na forma de módulos (ou populações) distribuídos de forma hierárquica permite a implementação de um processo mais elaborado na escolha do ponto de mutação: a escolha baseada no *fitness*. Neste trabalho a definição de qual ponto será modificado por mutação poderá ser feita de duas maneiras:

- escolha aleatória do ponto de mutação (EAI_{pmut}): todos os pontos de todos os indivíduos têm a mesma probabilidade de serem escolhidos;
- escolha do ponto de mutação baseada no *fitness* (EFi_{pmut}): a codificação hierárquica permite que, em algumas espécies, os alelos possam ser escolhidos com base no *fitness* (os piores têm maior chance de serem alterados), já que estes alelos representam indivíduos em populações hierarquicamente inferiores e possuem medidas de desempenho (*fitness*) associadas. Por exemplo, considere a população de base de regras (nível III). Neste caso, todos os alelos de um indivíduo que representa a base de regras selecionada para sofrer mutação representam regras individuais no nível II que possuem medidas de desempenho associadas.

A decisão sobre qual a forma de escolha do ponto de mutação é feita a cada geração e de maneira aleatória:

$$\text{Escolha do ponto de mutação}(t) \begin{cases} EAI_{\text{pmut}} \text{ se } r_p \leq 0.5 ; \\ EFi_{\text{pmut}} \text{ se } r_p > 0.5 , \end{cases}$$

onde t define a geração atual e r_p é um número aleatório no intervalo $[0, 1]$. Desta forma, o processo evolutivo pode privilegiar uma busca mais direcionada em algumas gerações e manter o caráter aleatório da mutação em outras.

Conforme visto anteriormente, os alelos do nível I (partições nebulosas), representam valores relativos. Esta flexibilidade na codificação permite que qualquer ponto do cromossomo possa ser

escolhido no processo de crossover e mutação. A mutação para valores reais, utilizada no nível I, altera o formato e a localização (suporte) das funções de pertinência, mas restrições são impostas para garantir a geração de indivíduos válidos. Neste nível, as duas condições de visibilidade, γ -completude e κ -sobreposição, devem ser sempre observadas de modo a gerar partições nebulosas interpretáveis. Estas propriedades, que determinam graus de sobreposição mínimo (γ) e máximo (κ) entre as funções de pertinência no universo de cada variável de entrada, são garantidas pela restrição no domínio $D_k = [l_k, u_k]$ associado a cada alelo no *locus* k (ver capítulo 2, seção 2.3.3). Para o crossover, se as condições de visibilidade forem violadas, um procedimento de reparação é utilizado para realocar as funções de pertinência para o limite mais próximo das posições válidas. Esta reparação é feita no fenótipo com consequência para o genótipo de cada indivíduo alterado. Os algoritmos de reparação representam uma grande vantagem das técnicas evolutivas no ajuste das funções de pertinência, quando comparadas com técnicas de otimização clássicas. Usualmente, problemas de otimização restrita vão sempre requerer procedimentos de solução mais elaborados (Bäck *et al.*, 2000a,b).

No segundo nível (regras individuais), crossover e mutação são aplicados de modo a obter diferentes combinações dos termos lingüísticos. A mutação, aplicada aos cromossomos de valores inteiros, altera o valor atual v_k^t para um novo valor escolhido no alfabeto $\text{Alf}_k = \{0, 1, \dots, G_k\} - v_k^t$, onde G_k identifica o número máximo de termos lingüísticos (granularidade máxima) para a k -ésima variável de entrada, e o valor nulo codifica a irrelevância dessa variável para a respectiva regra. É importante salientar que, além de possibilitar o surgimento de regras mais simples, as condições irrelevantes imprimem uma outra característica à abordagem que está sendo proposta: a seleção de variáveis ou atributos (Hall, 1998). Isto porque, se numa determinada geração t , a k -ésima variável é irrelevante em todas as regras individuais R_j que compõem uma determinada base de regras b , esta variável é considerada irrelevante na aplicação, para todos os sistemas nebulosos que utilizam a base de regras b . Isto ocorre quando $v_{jk}^t = 0 \quad \forall j = 1, \dots, m$, onde m representa o tamanho da base de regras b .

No nível das bases de regras (nível III), crossover e mutação alteram os índices inteiros associados às regras individuais. O valor atual v_k^t é alterado por mutação para um novo valor, escolhido no alfabeto $\text{Alf}_k = \{0, 1, \dots, S_{II}\} - v_k^t$, onde S_{II} identifica o tamanho da população do nível II, ou seja o total de regras individuais possíveis. O valor nulo indica a eliminação da regra. Neste nível, caso a escolha do ponto de mutação seja do tipo EFi_{pmut} , alelos que representam as regras individuais com baixo *fitness* têm maior probabilidade de serem alterados por mutação.

No nível dos sistemas nebulosos (nível IV), no caso de adoção de modelos TS, apenas os ope-

radores de agregação dos antecedentes (\mathbf{A}_a), índices que apontam para a base de regras (BR) e partições nebulosas (PN) são considerados. O operador de crossover produz uma recombinação de dois cromossomos testando diferentes operadores aplicados às diferentes bases de regras e partições nebulosas. A mutação inteira, aplicada aos alelos com valores inteiros, altera os valores antigos para novos, escolhidos dentre todas as possibilidades. Por exemplo, os novos valores dos alelos localizados no *locus* 2 (v_2^t), que identificam o operador de agregação dos antecedentes (\mathbf{A}_a) (ver figura 4.3), são escolhidos do conjunto $\{1, \dots, 9\} - v_2^t$, onde os índices inteiros identificam as t-normas apresentadas na seção 3.1.3. Para os alelos no *locus* 8 (v_8^t), que identificam a base de regras (BR), os novos valores são escolhidos do conjunto, $\{0, 1, \dots, S_{III}\} - v_8^t$ e para os alelos no *locus* 9 (v_9^t), que identificam as partições nebulosas (PN), são selecionados dentre os possíveis valores do conjunto $\{0, 1, \dots, S_I\} - v_9^t$, onde S_{III} e S_I identificam os tamanhos das populações no nível III (base de regras) e I (partições nebulosas), respectivamente. O operador de mutação real é aplicado aos alelos dos *loci* 3 e 5, que estão associados aos parâmetros p_i . Para os modelos Mamdani, diferentes operadores de agregação das regras (\mathbf{A}_R) e semântica das regras (f), assim como alternativas distintas para os métodos de transformação da saída nebulosa em não-nebulosa (MoM, CoA, CoG), são testados através de modificações realizadas pelos operadores genéticos.

Uma vez que o elitismo é adotado, deve-se garantir que as operações realizadas nos níveis inferiores não irão degenerar o melhor sistema nebuloso, codificado no nível IV. Daí, para os níveis I, II e III, após a execução de todos os operadores evolutivos aplicados a cada nível, o melhor sistema nebuloso é avaliado e as operações genéticas do nível atual que pioram o desempenho do melhor indivíduo no nível IV são descartadas. Este procedimento que descarta as operações no cromossomo (em um nível hierárquico inferior) determina a substituição deste cromossomo pelo cromossomo correspondente na geração anterior.

4.3.5 Avaliação do *Fitness*

No processo de avaliação do *fitness* de indivíduos pertencentes a diferentes espécies, relações intermodulares são estabelecidas: o *fitness* de um indivíduo é calculado levando-se em conta o *fitness* de indivíduos pertencentes a outras populações. A figura 4.8 ilustra o esquema geral de avaliação do *fitness* adotado pelo processo co-evolutivo, para o qual a avaliação nas diferentes espécies é dada por:

1. Sistema Nebuloso (nível IV): $F(\text{SN}_i^t)$ baseado no desempenho do respectivo sistema no domínio

da aplicação;

2. Base de Regras (nível III): $F(\text{BR}_k^t) = \max (F(\text{SN}_b^t), \dots, F(\text{SN}_d^t))$; onde b, \dots, d representam os sistemas nebulosos que utilizam a base de regras k ;
3. Regra Individual (nível II): $F(\text{RI}_j^t) = \text{média} (F(\text{BR}_m^t), \dots, F(\text{BR}_n^t))$, onde m, \dots, n identificam as bases de regras que utilizam a regra individual j ;
4. Partição Nebulosa (nível I): $F(\text{PN}_q^t) = \max (F(\text{SN}_x^t), \dots, F(\text{SN}_z^t))$, onde x, \dots, z representam os sistemas nebulosos que utilizam a partição nebulosa q .

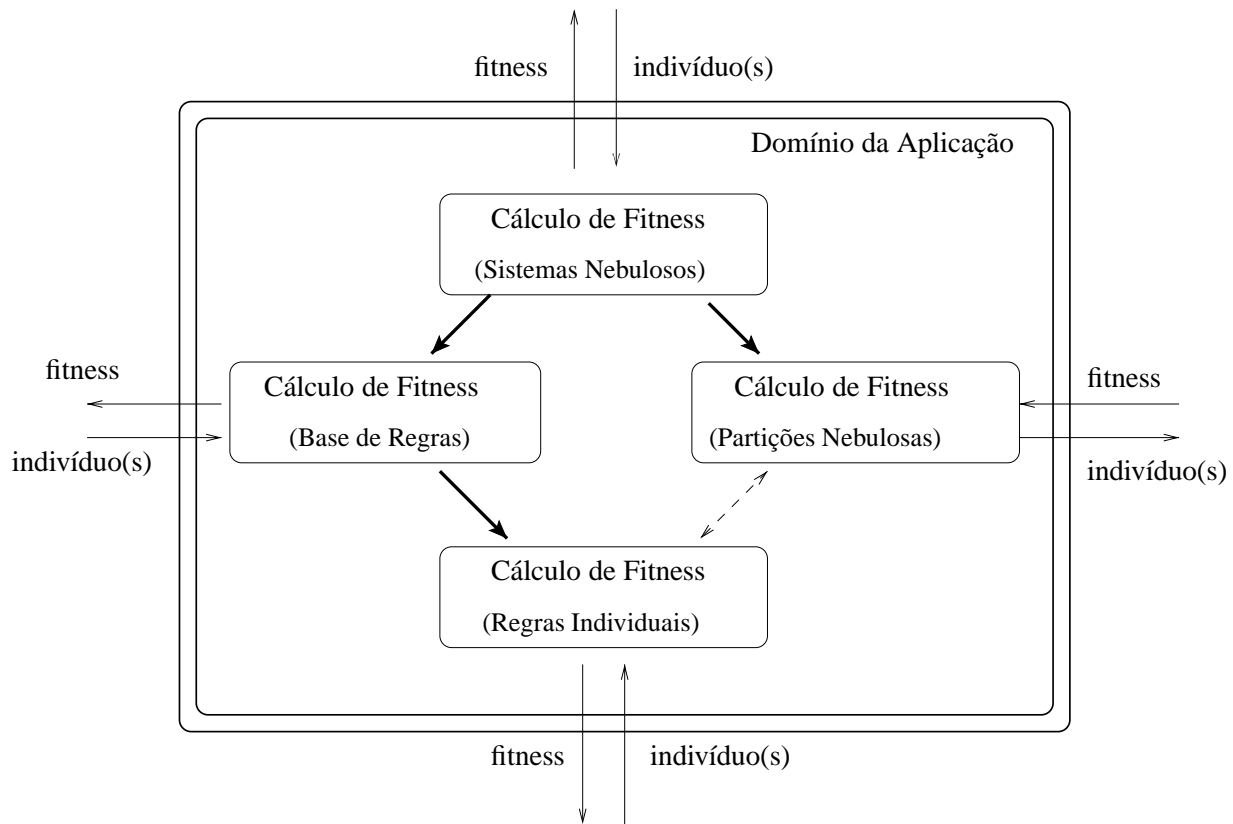


Figura 4.8: Esquema de avaliação do *fitness*.

No nível IV, o *fitness* de cada indivíduo é avaliado decodificando-se o cromossomo que representa um sistema nebuloso e medindo o seu desempenho na solução do problema que está sendo considerado. Por exemplo, para problemas de aproximação, em geral o *fitness* de cada sistema nebuloso é computado como

$$\text{fitness} = \frac{1}{\sqrt{\text{EQM}}},$$

onde o EQM é dado pela equação (4.2), seção 4.1.

O *fitness* de cada indivíduo dos níveis I e III é calculado baseado no desempenho dos sistemas nebulosos nos quais ele participa. Neste caso, o melhor sistema nebuloso, escolhido dentro de um grupo de indivíduos no nível IV que utilizam a mesma BR e PN, define o valor do *fitness* de sua base de regras e partição nebulosa. Isto se justifica pelo fato de que, se uma determinada base de regras ou partição nebulosa está presente em um sistema nebuloso com bom desempenho, o seu *fitness* não deve ser piorado devido à sua participação em um sistemas nebulosos de *fitness* baixo, daí a razão de se usar o operador max. Para as regras individuais (nível II), uma vez que existe uma grande interação delas e uma regra sozinha dificilmente responde pelo *fitness* do sistema nebuloso, a média é mais indicada. Neste caso, a regra é avaliada pela medida do quanto ela coopera com outras regras para compor um sistema nebuloso no nível IV. A seta tracejada presente na figura 4.8 indica uma possível relação, no cálculo do *fitness*, entre os indivíduos das duas espécies envolvidas (partições nebulosas e regras individuais). No entanto, neste trabalho nenhuma relação (no cálculo do *fitness*) entre estas duas espécies será considerada.

4.4 O Modelo do Sistema Nebuloso com Conseqüentes Não-Lineares

O processo co-evolutivo descrito anteriormente permite a utilização de um dentre os dois modelos de sistemas nebulosos mais comuns: Mamdani e Takagi-Sugeno. Na etapa de simulações e resultados mostrados no capítulo 5, a ênfase será dada para os modelos TS pois, apesar da menor interpretabilidade lingüística, estes modelos são extremamente úteis na representação de sistemas complexos com um número reduzido de regras (melhorando a característica de compactação) e permitem o uso de algoritmos de estimação numérica, extremamente eficientes para se obterem os parâmetros dos conseqüentes (melhorando a acuidade do modelo). Para maiores detalhes sobre as aplicações envolvendo o modelo Mamdani, ver Delgado *et al.* (1999).

Nos modelos TS, o conseqüente de cada regra pode ser formulado diretamente por uma relação funcional paramétrica entre as variáveis de entrada. Como exemplo, seja o sistema nebuloso composto por m regras nebulosas $R_j, j = 1, \dots, m$, na forma:

R_j : Se X_1 é A_1^j E X_2 é A_2^j E ... E X_n é A_n^j então Y é $g_j(\mathbf{w}_j, \mathbf{x})$,

onde $\mathbf{x} = [x_1 \cdots x_n]$ representa o vetor de variáveis base de entrada, e o vetor \mathbf{w}_j (de dimensão Q) contém os parâmetros da função do conseqüente $g_j(\cdot)$. Os tipos mais comuns encontrados na literatura são as funções constante ($Q = 1$) e linear ($Q = n + 1$). Aqui, funções não-lineares ($Q = \frac{n(n-1)}{2} + 2n + 1$) serão também consideradas, como a mostrada a seguir:

$$g_j(\mathbf{w}_j, \mathbf{x}) = w_{j0} + w_{j1}x_1 + \cdots + w_{jn}x_n + w_{j(n+1)}x_1x_1 + \cdots + w_{j(2n)}x_1x_n + w_{j(2n+1)}x_2x_2 + \cdots + w_{j(\frac{n(n-1)}{2}+2n)}x_nx_n. \quad (4.3)$$

A inclusão de outros termos na equação 4.3 depende da disponibilidade de recursos computacionais. Por exemplo, para problemas com mais de duas variáveis de entrada (como no caso dos dados da IRIS apresentados na seção 5.4.2), o termo envolvendo o produto cruzado de todas as variáveis pode ser incluído de modo aumentar ainda mais a flexibilidade do modelo.

A base de regras do modelo TS utilizado ao longo deste trabalho pode ser de dois tipos:

- base de regras específicas (BRE): utiliza somente regras específicas nas quais todas as variáveis são relevantes (na codificação, $A_k^j > 0$);
- base de regras genéricas (BRG): utiliza também as regras genéricas nas quais existem variáveis irrelevantes (na codificação, $A_k^j \geq 0$).

A estrutura e os parâmetros do modelo TS são determinados pelo algoritmo co-evolutivo descrito na seção 4.3. Após a especificação destes parâmetros, os elementos do vetor $\mathbf{w}_j = [w_{j0} \cdots w_{jQ-1}]^T$ podem ser calculados utilizando-se as técnicas de otimização que serão discutidas nas seções 4.4.2 e 4.4.3.

4.4.1 Exemplo de Utilização do Modelo TS com Conseqüentes Não-Lineares

Suponha um sistema nebuloso TS com conseqüentes não-lineares obtido ao final do processo co-evolutivo pela abordagem descrita na seção 4.3, atuando num domínio de aplicação envolvendo duas variáveis x_1 e x_2 e composto por 5 regras conforme mostrado na tabela 4.2.

Suponha também que o operador de agregação do antecedente obtido (\mathbf{A}_a) tenha sido o **min** e que as partições nebulosas nos universos das variáveis base x_1 e x_2 definem as FP associadas aos termos lingüísticos *baixo*, *médio*, e *alto* como ilustrado na figura 4.9.

Então, o mapeamento de entrada-saída do sistema nebuloso que utiliza a base de regras mostrada na tabela 4.2, definido numa região compacta $\Omega \equiv [0, 1]^2$, e com partição nebulosa ilustrada na figura 4.9, seria como o mostrado na figura 4.10(a). As figuras 4.10(b)-(d) ilustram a influência de

Tabela 4.2: Parâmetros do modelo TS exemplificado

Regra	A_1^j	A_2^j	w_{j0}	w_{j1}	w_{j2}	w_{j3}	w_{j4}	w_{j5}
1	baixo	baixo	-1	0	0	0	0	0
2	baixo	alto	1	0	0	-0.19	0.47	0
3	médio	médio	-0.43	-0.37	-0.35	0.62	-0.77	0.5
4	alto	baixo	2	-0.5	0	0	0	0.85
5	alto	alto	1	0.13	0.22	0	0	0

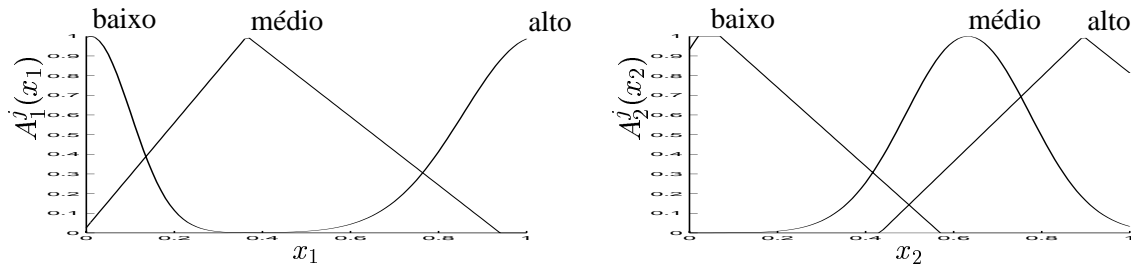


Figura 4.9: Partição nebulosa utilizada pelo modelo TS exemplificado.

cada regra agindo de forma isolada e auxiliam no entendimento do modelo TS com conseqüentes não-lineares, a ser adotado neste trabalho. O exemplo de comportamento entrada-saída mostrado na figura 4.10 procura evidenciar a capacidade de aproximação de mapeamentos de entrada-saída não-lineares por parte de sistemas nebulosos do tipo TS e com conseqüentes não-lineares. Este exemplo recorreu a um mapeamento do \mathbb{R}^2 para o \mathbb{R}^1 , portanto um mapeamento tridimensional, por questões de visualização gráfica, devendo-se salientar a validade das conclusões extraídas para mapeamentos de maior dimensão. Nos problemas de aproximação de funções e classificação de padrões, a serem considerados no capítulo 5, o sistema nebuloso que irá implementar o mapeamento entrada-saída, possivelmente não-linear, deverá ser obtido automaticamente a partir de dados de entrada-saída disponíveis. Os parâmetros do sistema nebuloso serão otimizados por co-evolução e, no caso de modelos TS, será necessário também otimizar os parâmetros da função que representa o conseqüente de cada regra. Não é conveniente obter estes parâmetros também por processos evolutivos, pois a relação deles com a saída do sistema nebuloso é linear (ver seção 4.4.2), o que permite a aplicação de poderosos métodos de otimização, caracterizados por fornecer a solução em uma única expressão algébrica, portanto de forma fechada.

4.4.2 Otimização do Conseqüente: Método Global

No processo de estimação global dos conseqüentes TS, todas as regras nebulosas são consideradas simultaneamente e o método dos quadrados mínimos é utilizado para se encontrar o conjunto ótimo

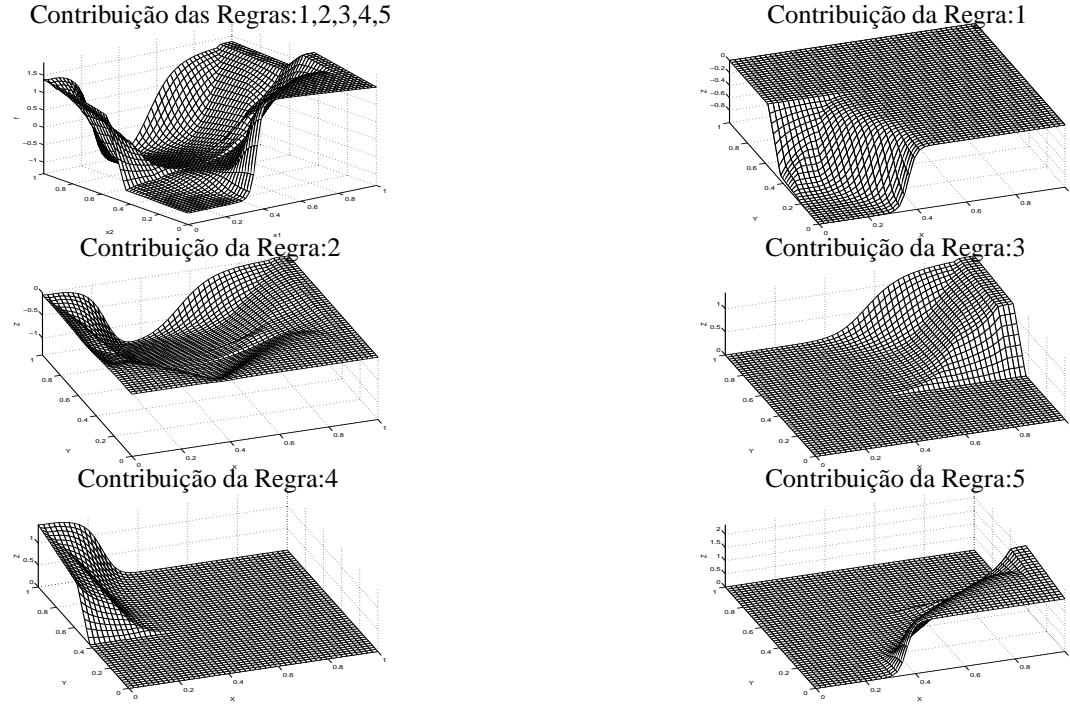


Figura 4.10: Mapeamento Produzido pelo Modelo TS exemplificado, considerando-se primeiramente a contribuição de todas as regras e depois a contribuição de cada uma delas separadamente.

de parâmetros $\mathbf{w}^* = [(\mathbf{w}_1^*)^T \cdots (\mathbf{w}_m^*)^T]^T$, onde $\mathbf{w}^* \in \mathfrak{R}^{mQ}$ e m define o total de regras.

Sejam n variáveis de entrada e $\mu_j(\mathbf{x}_p)$ o resultado da agregação do antecedente da j -ésima regra nebulosa para o p -ésimo padrão de entrada $\mathbf{x}_p = [x_{1p} \cdots x_{np}]$, $p = 1, \dots, N$. Então, a p -ésima saída do sistema nebuloso é dada por

$$y(\mathbf{x}_p) = \frac{\sum_{j=1}^m \mu_j(\mathbf{x}_p) g_j(\mathbf{w}_j, \mathbf{x}_p)}{\sum_{j=1}^m \mu_j(\mathbf{x}_p)}. \quad (4.4)$$

Sabendo-se que $g_j(\mathbf{w}_j, \mathbf{x}_p)$ é dado pela equação (4.3), então é possível definir β_j como

$$\beta_j(\mathbf{x}_p) = \beta_j^p = \frac{\mu_j(\mathbf{x}_p)}{\sum_{j=1}^m \mu_j(\mathbf{x}_p)}$$

e o vetor $\lambda_j(\mathbf{x}_p) \in \mathfrak{R}^Q$ como

$$\lambda_j(\mathbf{x}_p) = [\beta_j^p \quad \beta_j^p x_{1p} \quad \cdots \quad \beta_j^p x_{np} \quad \beta_j^p x_{1p} x_{1p} \quad \beta_j^p x_{2p} x_{2p} \quad \cdots \quad \beta_j^p x_{np} x_{np} \quad \beta_j^p x_{1p} \cdots x_{np}]^T,$$

de modo que a equação (4.4) possa ser reescrita como

$$y(\mathbf{x}_p) = (\lambda(\mathbf{x}_p))^T \mathbf{w},$$

onde $\mathbf{w} = [\mathbf{w}_1^T \cdots \mathbf{w}_m^T]^T$, e $\lambda(\mathbf{x}_p) \in \mathfrak{R}^{mQ}$ é dado por

$$\lambda(\mathbf{x}_p) = [\lambda_1^T(\mathbf{x}_p) \cdots \lambda_m^T(\mathbf{x}_p)]^T.$$

Agora, seja a matriz $\Lambda \in \mathfrak{R}^{N \times mQ}$ definida como

$$\Lambda = [\lambda(\mathbf{x}_1) \quad \lambda(\mathbf{x}_2) \quad \cdots \quad \lambda(\mathbf{x}_N)]^T.$$

Então, a saída estimada $\hat{\mathbf{y}} = [y(\mathbf{x}_1) \cdots y(\mathbf{x}_N)]^T \in \mathfrak{R}^N$ é dada por

$$\hat{\mathbf{y}} = \Lambda \mathbf{w}.$$

Este é um sistema linear de equações, demonstrando que a saída do sistema nebuloso está efetivamente linearmente relacionada com os parâmetros do conseqüente das regras TS. Além disso, como normalmente o número N de dados de entrada-saída disponíveis é maior do que o número mQ de parâmetros a serem otimizados, então resulta um sistema linear sobredeterminado. Considerando-se \mathbf{y}_d a saída desejada, encontra-se \mathbf{w}^* , através da solução do problema de otimização dos quadrados mínimos $\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{y}_d - \hat{\mathbf{y}}\|_2^2$, onde $\|\cdot\|_2$ representa a norma Euclidiana. Portanto,

$$\mathbf{w}^* = (\Lambda^T \Lambda)^{-1} \Lambda^T \mathbf{y}_d. \quad (4.5)$$

Como apontado por Golub & Van Loan (1996), \mathbf{w}^* produz a solução $\hat{\mathbf{y}}^* = \Lambda \mathbf{w}^*$, mas como $\text{posto}(\Lambda) = mQ < N$, então, embora seja válida a seguinte expressão:

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \left[\frac{1}{2} \|\hat{\mathbf{y}} - \mathbf{y}_d\|_2^2 \right],$$

é possível que $\hat{\mathbf{y}}^* \neq \mathbf{y}_d$, ou seja, o que se obtém é a solução mais próxima possível daquela desejada, em termos da norma Euclidiana.

Um outro aspecto da solução $\mathbf{w}^* \in \mathfrak{R}^{mQ}$ diz respeito à escolha de funções mais complexas. A introdução da não-linearidade no conseqüente resulta em um alto número Q de parâmetros a serem otimizados em cada uma das m regras. Se Q é maior do que o necessário em uma dada aplicação, alguns parâmetros devem ser “podados” para se evitar redundância nos conseqüentes das regras nebulosas. Portanto, para se garantir que a matriz $\Lambda^T \Lambda$ seja bem condicionada, um procedimento de poda deve ser aplicado (Delgado *et al.*, 2000b). O algoritmo que realiza este procedimento será detalhado na seção 4.4.4.

4.4.3 Otimização do Conseqüente: Método Local

O processo de estimação local otimiza o conseqüente considerando cada regra independentemente da outra (Hoffmann & Nelles, 2000; Yen *et al.*, 1998). Portanto, para N padrões de entrada e, considerando apenas a regra j , a saída do sistema nebuloso é

$$\hat{\mathbf{y}} = \hat{\mathbf{y}}_j = \Lambda_j \mathbf{w}_j ,$$

onde $\Lambda_j \in \Re^{N \times Q}$ é dado por

$$\Lambda_j = [\lambda_j(\mathbf{x}_1) \ \lambda_j(\mathbf{x}_2) \ \cdots \ \lambda_j(\mathbf{x}_N)]^T .$$

Neste caso, o conseqüente ótimo $\mathbf{w}_j^* = [w_{j0}^* \cdots w_{j(\frac{n(n-1)}{2} + 2n)}^*]^T$ pode ser obtido pelo método dos quadrados mínimos ponderados, formulado da seguinte forma:

$$\min_{\mathbf{w}_j} \frac{1}{2} \|\hat{\mathbf{y}} - \mathbf{y}_d\|_{\Psi_j}^2 = \min_{\mathbf{w}_j} \frac{1}{2} \left[(\hat{\mathbf{y}} - \mathbf{y}_d)^T \Psi_j (\hat{\mathbf{y}} - \mathbf{y}_d) \right] ,$$

onde Ψ_j é a matriz diagonal definida por:

$$\Psi_j = \begin{bmatrix} \beta_j^1 & 0 & 0 & \cdots & 0 \\ 0 & \beta_j^2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \beta_j^N \end{bmatrix} .$$

Portanto, o conjunto de parâmetros ótimos para a regra R_j é dado por:

$$\mathbf{w}_j^* = (\Lambda_j^T \Psi_j \Lambda_j)^{-1} \Lambda_j^T \Psi_j \mathbf{y}_d . \quad (4.6)$$

Como no caso global, se a matriz $\Lambda_j^T \Psi_j \Lambda_j$ for mal condicionada, um algoritmo de poda, como o descrito na seção a seguir, deve ser aplicado.

4.4.4 O Processo de Poda

A hipótese de que a matriz $\Lambda \in \Re^{N \times mQ}$ seja de posto completo e que $N \geq \frac{n(n-1)}{2} + 2n + 1$ no método local e, $N \geq m \left(\frac{n(n-1)}{2} + 2n + 1 \right)$ no método global, são condições necessárias e suficientes para que \mathbf{w}^* seja obtido de forma fechada. A primeira condição pode ser sempre satisfeita porque, nos casos em que isto não acontece, eliminam-se as colunas linearmente dependentes (LD), até que a matriz Λ tenha posto completo.

A seguir, será apresentado o algoritmo que elimina as colunas LD de Λ , sempre que a matriz $\Lambda^T \Lambda$ for mal condicionada (Delgado *et al.*, 2000b). Este algoritmo se baseia no número de condição de uma matriz Z obtido pelo estimador de condição LAPACK (Manual Matlab, 1999) e que apresenta as seguintes características:

- o estimador calcula a condição recíproca de Z na norma simples: $\text{RCOND}(Z)$;
- o número de condição $\text{RCOND}(Z)$ descreve o quão sensível é a matriz Z ao erro do vetor do lado direito, quando sistemas de equações são solucionados;
- $\text{RCOND}(Z)$ se aproxima de 1 quando Z é bem-condicionada
- $\text{RCOND}(Z)$ se aproxima de 0 quando Z é mal-condicionada.

O processo de poda proposto aumenta o número de condição recíproca RCOND da matriz $\Lambda^T \Lambda$, eliminando as colunas de Λ que mais contribuem para a redução de $\text{RCOND}(\Lambda^T \Lambda)$.

Para o caso local, a eliminação das colunas LD é feita na matriz Λ_j , com conseqüência no $\text{RCOND}(\Lambda_j^T \Psi_j \Lambda_j)$, $j = 1, \dots, m$.

Os passos principais do processo de poda estão detalhados no fluxograma ilustrado na figura 4.11. O algoritmo de poda atua tentando produzir uma matriz cujas colunas sejam todas linearmente independentes (LI) e cujo número de condição RCOND seja superior a um limiar pré-estabelecido L_{prun} . Para isso, recebe como entrada a matriz $Z = \Lambda(\Phi)$, onde $\Phi =$ colunas LI+LD identifica o conjunto de todos os índices das colunas de Λ , e produz na saída, a matriz $Z(\mathbf{r})$, onde $\mathbf{r} =$ colunas LI identifica as colunas de Z que devem ser mantidas. O algoritmo inicia considerando a matriz $Z_1 = Z(\mathbf{p})$, onde \mathbf{p} representa o conjunto de índices das colunas não-nulas. Caso a matriz Z_1 possua número de condição superior ao limiar L_{prun} , esta matriz é escolhida como a matriz final. Caso o seu número de condição esteja abaixo do limiar L_{prun} , todas as suas colunas têm seu status atualizado para o valor 0.5, indicando que estas colunas serão submetidas a teste. A partir daí inicia-se um processo de busca pelo conjunto \mathbf{r} de colunas que produzam o melhor número de condição na matriz $Z_r^T Z_r$, começando com a coluna C_i de maior norma Euclidiana $N_{\text{Ecl}}(C_i)$. Caso o número de condição de $Z_2 = C_i^T C_i$ seja superior ao limite mínimo, esta coluna é escolhida e seu status é atualizado para 1. Realiza-se então uma busca entre todas as colunas que ainda não foram escolhidas e seleciona-se como candidata aquela que produza a melhor combinação (em termos do número de condição) quando agregada à matriz atual Z_2 . Esta nova coluna será escolhida caso o número de condição da nova matriz Z_3 fique acima do limiar L_{prun} mínimo e, neste caso, terá seu status atualizado para o valor 1. Este processo de teste com as colunas se repete até que a agregação

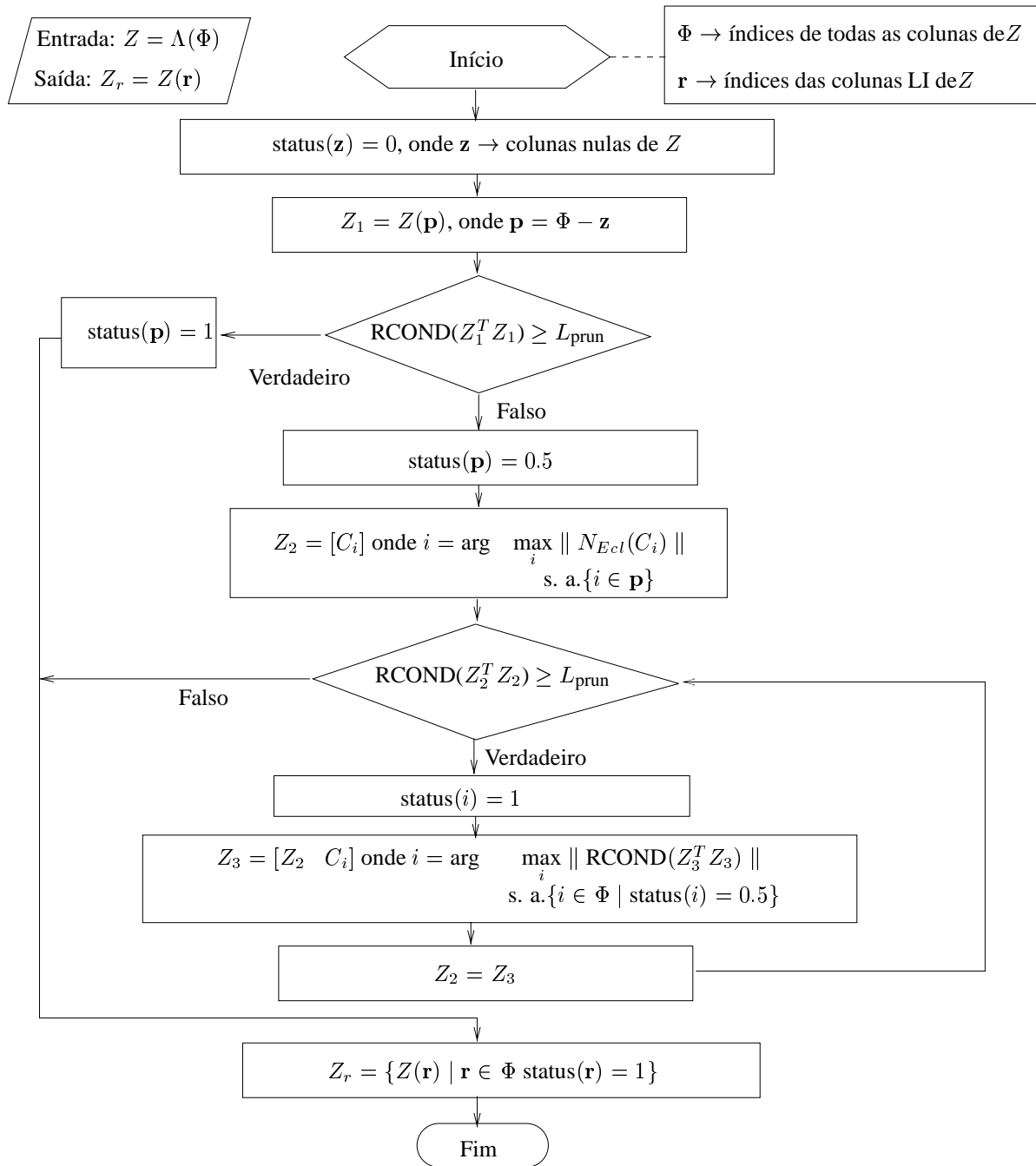


Figura 4.11: Algoritmo para o processo de poda.

de uma nova coluna leve o número de condição para um valor abaixo do limiar L_{prun} . Nesta caso, a última coluna candidata é descartada e todas as escolhidas anteriormente são selecionadas para compor a matriz final Z_r .

Uma vez que $Z_r = \Lambda_r \in \mathfrak{R}^{N \times mQ'}$ é de posto completo o vetor de parâmetros reduzido \mathbf{w}_r^* é determinado através da equação:

$$\mathbf{w}_r^* = (\Lambda_r^T \Lambda_r)^{-1} \Lambda_r^T \mathbf{y}_d .$$

O novo conjunto de parâmetros \mathbf{w}_r^* é um subconjunto do original \mathbf{w}^* . A cardinalidade deste novo conjunto depende do limiar escolhido L_{prun} . Os parâmetros que pertencem a \mathbf{w}^* , mas não pertencem a \mathbf{w}_r^* , i.e., aqueles cujas colunas $\text{LD}=\Phi - \mathbf{r}$ receberam status inferior a 1, são definidos com valor nulo, indicando a redundância destes na regra correspondente. A eliminação de todos os parâmetros do conseqüente de uma determinada regra, indica a redundância desta regra na base de regras do sistema nebuloso correspondente.

No método de otimização global, a dimensão da matriz $\Lambda_r^T \Lambda_r$ a ser invertida cresce de forma quadrática com o número n de variáveis de entrada. Para problemas com muitas variáveis envolvidas (valor de n alto), pode ser mais apropriado diminuir a complexidade da função do conseqüente, reduzindo-se assim a dimensão de \mathbf{w} para aliviar o custo computacional.

Para o caso local, a matriz $Z_r = \Lambda_{jr} \in \mathfrak{R}^{N \times Q'}$ é de posto completo e o vetor de parâmetros reduzido \mathbf{w}_{jr}^* é determinado através da equação:

$$\mathbf{w}_{jr}^* = (\Lambda_{jr}^T \Psi_j \Lambda_{jr})^{-1} \Lambda_{jr}^T \Psi_j \mathbf{y}_d .$$

O processo de poda, ao lado dos operadores genéticos, melhora a acuidade do modelo nebuloso, pois possibilita o cálculo do conseqüente de forma fechada. Além disso, aumenta a interpretabilidade do modelo TS através da obtenção de bases de regras mais compactas e formadas por regras nebulosas consistentes, nas quais os termos redundantes não estão presentes.

Capítulo 5

Simulações e Resultados

O objetivo deste capítulo é testar, em diversos domínios de aplicação, os sistemas nebulosos resultantes da abordagem co-evolutiva proposta no capítulo 4. Além da acuidade, será analisada a interpretabilidade dos modelos nebulosos obtidos, medida com base nos critérios de visibilidade, simplicidade, compactação e consistência. A característica de autonomia de projeto ficará evidente uma vez que os resultados enfatizarão os parâmetros críticos dos modelos TS, como: os operadores de agregação dos antecedentes; número e tipo das regras nebulosas; formatos e localização das funções de pertinência. Todos eles são obtidos de forma automática pela abordagem co-evolutiva.

Este capítulo traz os resultados das simulações realizadas para diversos problemas. Na seção 5.1, são apresentadas outras abordagens que deverão ser testadas na etapa de simulações e que se baseiam em técnicas distintas da computação flexível. Esta seção traz ainda as considerações gerais sobre cada uma destas abordagens. A seção 5.2 mostra os resultados da aproximação de duas funções no \mathfrak{R}^3 para dados com alto grau de confiabilidade. Esta seção traz duas análises distintas: no primeiro caso, é feita uma análise de acuidade levando-se em conta a capacidade de generalização dos modelos obtidos e custos computacionais envolvidos; no segundo caso, embora o objetivo também seja minimizar o erro de aproximação, os modelos nebulosos obtidos são avaliados em termos da interpretabilidade, medida principalmente pelos critérios de visibilidade e compactação. A seção 5.3 traz problemas de aproximação no \mathfrak{R}^2 e no \mathfrak{R}^3 , sendo que, em ambos os casos, os dados disponíveis para treinamento são perturbados por ruído branco com distribuição normal. Na seção 5.4, os sistemas nebulosos são aplicados a dois problemas de classificação distintos: classificação no \mathfrak{R}^2 , utilizando os dados das espirais concêntricas, e classificação no \mathfrak{R}^4 , utilizando os dados da IRIS (Fisher, 1936). No problema da IRIS, uma análise de custo computacional será novamente realizada. Por fim, na seção 5.5, um experimento denominado de experimento de caminho

fechado é realizado com o objetivo de avaliar a capacidade de busca do sistema co-evolutivo. Nesta seção, considera-se que o mapeamento no \mathcal{R}^3 a ser aproximado é produzido por um sistema nebuloso específico, e todas as simulações são realizadas considerando-se diferentes níveis de dificuldade no espaço de busca, variando do caso mais simples para o mais complexo.

5.1 Abordagens de Comparação

Nesta etapa de simulações e apresentação dos resultados, os modelos nebulosos resultantes da abordagem co-evolutiva (denominada de CoevolGFS - *Coevolutionary Genetic Fuzzy Systems*) serão comparados com outros sistemas baseados em Computação Flexível (CF).

- Com o objetivo de avaliar os benefícios resultantes do uso da co-evolução, hierarquia e cooperação entre os indivíduos de diferentes espécies, os sistemas nebulosos resultantes da abordagem proposta serão comparados com os modelos obtidos por uma abordagem evolutiva tradicional implementada especialmente para esta comparação e denominada de EGFS - *Evolutionary Genetic Fuzzy Systems*. Esta abordagem apresenta as seguintes características:
 - evolui modelos TS com conseqüentes não-lineares;
 - o operador de agregação dos antecedentes é fixado pela abordagem como o **min**;
 - evolui a base de regras, definindo quantas e quais regras irão compor o sistema nebuloso;
 - ajusta os parâmetros das funções de pertinência dos universos das variáveis de entrada, que podem assumir três tipos: triangular, trapezoidal e Gaussiana, e cuja codificação (baseada em posições relativas) é a mesma adotada pela abordagem co-evolutiva;
 - os parâmetros da base de regras e os relacionados com as funções de pertinência são codificados no mesmo cromossomo. O conjunto destes cromossomos irá formar uma única população a ser evoluída ao longo das gerações.

A figura 5.1 ilustra a codificação adotada pelo modelo EGFS. O cromossomo (ver figura 5.1(a)) é formado por duas partes: base de regras (BR) e partições dos universos associados (PN). A codificação destas duas partes (concatenadas para formar o cromossomo) é análoga àquela utilizada pelos modelos co-evolutivos. Na parte do cromossomo que representa a base de regras, os alelos assumem valores inteiros, e cada regra (R_j) é associada a uma seqüência, representando índices dos termos lingüísticos nas proposições de entrada. Na

parte do cromossomo que representa as partições nebulosas (PN), uma seqüência de 5 alelos codifica cada função de pertinência FP. Primeiramente, são codificadas todas as FPs de cada variável, que só então são agrupadas para formar a partição final. Uma única população (ver figura 5.1(b)) evolui ao longo das gerações. A descrição dos operadores evolutivos adotados será apresentada na seção 5.1.1.

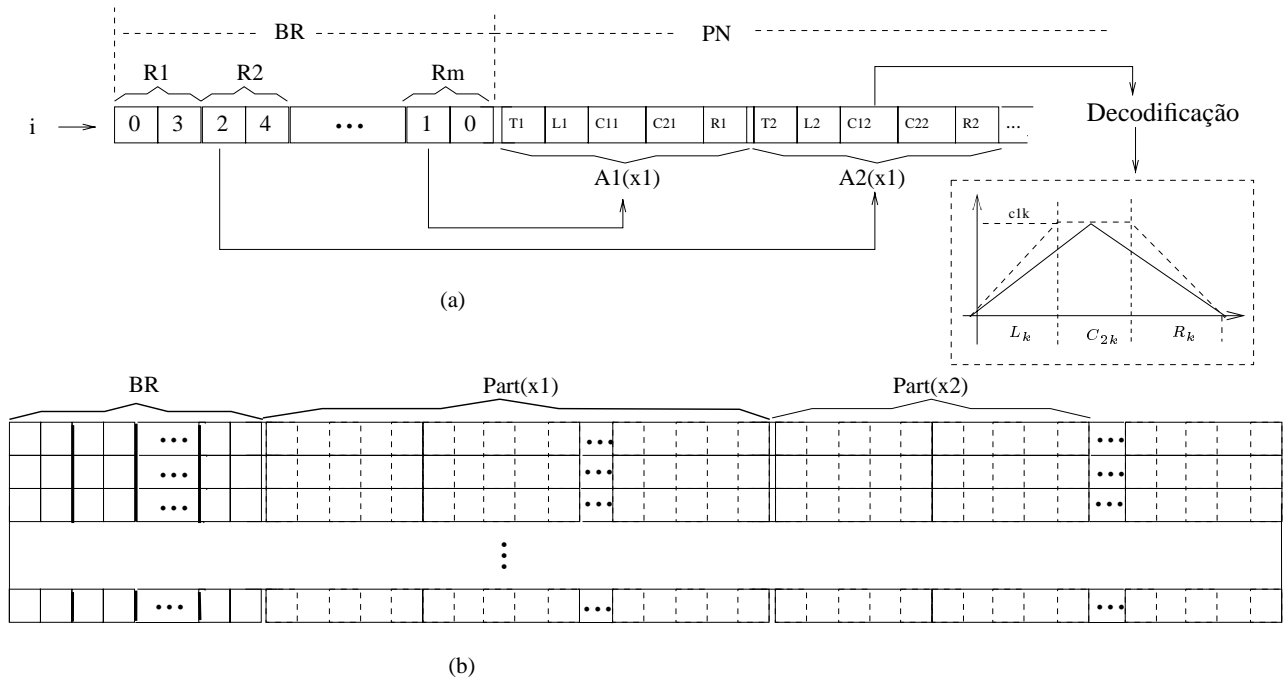


Figura 5.1: Modelo evolutivo de comparação (supondo duas variáveis de entrada): (a) indivíduo; (b) população.

- Visando a avaliação de outras técnicas da computação flexível na produção de sistemas híbridos, um sistema neuro-nebuloso será utilizado. Na maioria das simulações realizadas, os resultados do modelo co-evolutivo proposto serão comparados com aqueles obtidos pelo ANFIS que é, reconhecidamente, uma das referências mais importantes para comparar técnicas de projeto de sistemas nebulosos que usam aprendizado e adaptação (Nauck & Kruse, 1999; Rizzi *et al.*, 1999).
- Para avaliar o compromisso acuidade/interpretabilidade dos modelos resultantes, em algumas simulações utilizou-se uma rede neural multicamadas (Haykin, 1999), com uma única camada intermediária e treinada com *backpropagation*, denominada MLP-BckPr.

- Com o objetivo de avaliar os aspectos de generalização num contexto de classificação de padrões, o modelo obtido pela abordagem proposta será também comparado com o classificador nebuloso proposto por Castellano & Fanelli (2000). Este classificador, aqui denominado de *CompacFuzzClass*, é obtido por uma abordagem que opera em três etapas:
 1. primeiramente, as regras nebulosas são geradas por um processo de agrupamento adaptativo (*adaptively clustering*) que particiona o espaço de entrada gerando aglomerados (*clusters*) que representam as funções de pertinência, e define uma relação entre estas funções e os rótulos das classes;
 2. num segundo estágio, a acuidade do classificador é melhorada treinando uma rede neuro-nebulosa usada para modelar o classificador nebuloso;
 3. por fim, a interpretabilidade do classificador nebuloso é incrementada através da compressão da base de regras.

Na abordagem proposta por Castellano & Fanelli (2000) os dados de teste são utilizados para validar o modelo obtido na etapa 3.

5.1.1 Considerações Gerais Sobre as Abordagens de Comparação

Algumas considerações gerais serão feitas nesta etapa de simulações e apresentação dos resultados:

- **Tamanho fixo da(s) população(ões):**
 - CoevolGFS: As quatro populações representando as partições nebulosas (nível I), regras individuais (nível II), bases de regras (nível III), e sistemas nebuloso (nível IV), têm tamanho fixo $S_I, S_{II}, S_{III}, S_{IV}$, respectivamente;
 - EGFS: Uma única população de tamanho fixo é utilizada no processo evolutivo.
- **Fixação da população do nível II:** Em aplicações da abordagem CoevolGFS envolvendo um grande número de entradas e termos lingüísticos, a evolução da população de regras individuais (nível II) é necessária, em virtude do elevado número de proposições nebulosas possíveis e ocorre conforme descrito no algoritmo proposto na seção 4.3. Entretanto, como todas as simulações apresentadas neste capítulo envolvem um número relativamente pequeno de variáveis de entrada (no máximo 4), o nível II não sofre alterações ao longo do processo evolutivo. Nestes casos, o nível II é fixo e representado por um conjunto formado por todas

as proposições nebulosas possíveis, com cardinalidade dada por $S_{II} = S_{Esp} \cup S_{Gen}$, onde S_{Esp} fornece o tamanho do conjunto de todas as regras específicas possíveis e S_{Gen} identifica o total de elementos do conjunto de todas as regras genéricas possíveis. No caso dos testes que utilizam as bases de regras específicas, tem-se $S_{II} = S_{Esp}$.

- **Operadores Evolutivos:**

- CoevolGFS: vários métodos de seleção e operadores de crossover e mutação foram testados, resultando em um conjunto de operadores evolutivos mais adequados: o método de seleção combinando seleção por torneio e seleção por diversidade; o elitismo que assegura a sobrevivência do melhor indivíduo para a próxima geração; o crossover simples, com escolha aleatória do ponto de troca; e a mutação uniforme aplicada a todos os alelos que assumem valores reais;
- EGFS: os mesmos operadores evolutivos adotados para a abordagem CoevolGFS foram utilizados.

- **Crítérios de Visibilidade:** Na abordagem CoevolGFS, os parâmetros dos critérios (γ -complete e κ -sobreposição) são fixos, e definidos como $\gamma = 0.02$ e $\kappa = 0.95$. As escolhas destes valores tem por objetivo atender o critério de visibilidade das partições nebulosas, sem, no entanto, restringir demasiadamente o processo de busca do modelo co-evolutivo. Tais definições não se aplicam ao ANFIS e ao EGFS;

- **Iterações:** As gerações do processo evolutivo e co-evolutivo, e as épocas do processo de treinamento do ANFIS serão contabilizadas por uma variável única denominada ciclo.

- **Granularidade máxima:** Em todas as abordagens, a granularidade máxima (GranulMax) dos universos das variáveis de entrada (x_1, x_2, \dots, x_n) é definida como

$$GranulMax = [G_1, G_2, \dots, G_n],$$

onde $G_k, k = 1, \dots, n$ representa o número máximo de FPs permitido para cada variável de entrada. Este parâmetro deverá ser definido pelo usuário em todas as abordagens.

- **Distribuição inicial das funções de pertinência:** Em todas as abordagens, as partições nebulosas iniciais são uniformes.

- CoevolGFS e EGFS: os tipos das funções de pertinência (triangular, trapezoidal ou Gaussiana) nas partições nebulosas uniformes diferem aleatoriamente de um indivíduo para o outro;
 - ANFIS: o tipo é fixo e deve ser definido pelo usuário.
- **Regra \mathcal{M}**
 - regra que associa os índices inteiros nas regras individuais aos termos lingüísticos definidos pelo usuário. Esta regra é definida de forma arbitrária e aplicada *a posteriori*.
- **Base de regras:**
 - CoevolGFS: permite a utilização de uma dentre as duas base de regras possíveis: base de regras específicas (BRE) ou base de regras genéricas (BRG). Em ambos os casos, o número máximo de regras (MaxReg) em cada base de regras pode ser limitado pela granularidade máxima ou pode ser definido pelo usuário, com um valor menor;
 - EGFS: base de regras genéricas (BRG), com MaxReg limitado pela granularidade;
 - ANFIS: base de regras específicas (BRE), com total de regras = MaxReg fixo e definido pela granularidade.
- **Classes de funções no conseqüente TS:**
 - CoevolGFS: supõe uma dentre três classes possíveis: constantes, lineares e não-lineares. As funções não-lineares possuem, entre outros, termos quadráticos das variáveis de entrada e por isso são denominadas funções quadráticas;
 - EGFS: supõe apenas o uso de funções quadráticas;
 - ANFIS: supõe apenas o uso de funções lineares.
- **Estimação dos parâmetros do conseqüente TS:**
 - CoevolGFS: os parâmetros ótimos, para as diferentes classes de funções nos conseqüentes, podem ser obtidos de duas formas distintas: 1) método dos quadrados mínimos descrito no capítulo 4, equação (4.5), e neste caso, o processo será denominado de método global; 2) método dos quadrados mínimos ponderados descrito no capítulo 4, equação (4.6), denominado de método local. Em ambos os casos, o processo de poda é utilizado;

- EGFS: quadrados mínimos (método global), com poda;
 - ANFIS: quadrados mínimos recursivos.
- **Limiar L_{prun} do algoritmo de poda:** O limiar L_{prun} do algoritmo de poda (descrito no capítulo 4, seção 4.4.4) utilizado pelas abordagens CoevolGFS e EGFS foi fixado como $L_{\text{prun}} = 1 \cdot 10^{-10}$.
 - **Dados de treinamento e teste:** Para todas as abordagens, os dados de treinamento e teste são uniformemente distribuídos no universo das variáveis associadas. Os erros resultantes dos testes realizados ao longo do processo evolutivo, co-evolutivo e de treinamento fornecem subsídios importantes acerca da capacidade de generalização do modelo obtido, quando contrastados com os erros de treinamento. Entretanto, os dados de teste não são utilizados em nenhuma etapa do processo evolutivo tradicional, do processo co-evolutivo, ou do processo de treinamento do ANFIS. Como apenas os dados de treinamento são utilizados na escolha do melhor indivíduo e os dados de teste não alteram o ajuste dos parâmetros dos sistemas nebulosos ou não determinam a condição de parada, estes dados (teste) não são considerados dados de validação.
 - **Fitness:** Para as abordagens evolutivas e co-evolutivas, o *fitness* de cada sistema nebuloso é computado como $\text{fitness} = 1/\text{Erro}_{tr}$, com erro de treinamento dado por $\text{Erro}_{tr} = \sqrt{\text{EQM}}$, para os problemas de aproximação e $\text{Erro}_{tr} = \text{MissClass}$, para os problemas de classificação de padrões, onde EQM é calculado pela equação (4.2) e MissClass define o total de padrões classificados incorretamente.
 - **Condição de parada :** número máximo de gerações T ou critério de erro para os dados de treinamento.

5.2 Aproximação de Funções: Dados sem Ruído

Esta seção considera dois problemas de aproximação de funções para avaliar, em termos de acuidade e interpretabilidade, os sistemas nebulosos obtidos pelo processo co-evolutivo. Diferentes classes de funções no conseqüente serão avaliadas e os parâmetros associados serão estimados pelos métodos de otimização local e global. Bases de regras genéricas e específicas serão testadas, comparando-se a capacidade de generalização dos modelos obtidos.

5.2.1 Aproximação do Mapeamento F_1 : Análise de Acuidade

Neste experimento, o objetivo é aproximar uma combinação linear de 10 funções Gaussianas, como mostrado na equação a seguir:

$$F_1 : \Omega \rightarrow \mathfrak{R}, \text{ onde } f_1(x_1, x_2) = \sum_{q=1}^{10} \left(w_q \exp \left(\frac{-\sqrt{(x_1 - k_{1q})^2 + (x_2 - k_{2q})^2}}{v_q^2} \right) \right),$$

com

$$w = [1.4 \quad 1.2 \quad -1.0 \quad 0.7 \quad 1.8 \quad -2.0 \quad 2.5 \quad -1.0 \quad -0.4 \quad 1.9]$$

$$v = [1.4 \quad 2.2 \quad 1.6 \quad 1.2 \quad 2.4 \quad 3.0 \quad 2.4 \quad 3.4 \quad 2.6 \quad 1.8]$$

$$k_1 = [-3 \quad -3 \quad -2 \quad -1 \quad -1 \quad 1 \quad 2 \quad 2 \quad 3 \quad 3]$$

$$k_2 = [3 \quad -3 \quad 0 \quad 1 \quad -1 \quad 0 \quad 3 \quad -3 \quad 1 \quad -1].$$

Este mapeamento é particularmente complexo para ser aproximado porque, a despeito do fato das não-linearidades serem de natureza local, a extensão (raio de influência) de cada função Gaussiana não é a mesma, conforme pode ser observado na figura 5.2(a). Os dados de entrada $(x_{1p}, x_{2p}), p = 1, \dots, N$ são uniformemente distribuídos e restritos à região compacta $\Omega = [-4, 4]^2$. A saída da função $f(x_{1p}, x_{2p}), p = 1, \dots, N$, está limitada ao intervalo $[0, 1.5]$.

O total de indivíduos em cada espécie da abordagem co-evolutiva (CoevoIGFS) é dado por: 30 indivíduos na população de sistemas nebulosos ($S_{IV} = 30$), 80 indivíduos na população de base de regras ($S_{III} = 80$) e 15 indivíduos na população de partições nebulosas ($S_I = 15$). Os operadores de crossover e mutação são aplicados com probabilidade $P_C = 0.2$ e $P_M = 0.08$, respectivamente, para os 3 níveis (I, III e IV) que sofrem alterações. Estes valores de P_C e P_M foram escolhidos após inúmeras tentativas de melhorar o desempenho do processo co-evolutivo. A granularidade máxima (GranulMax) foi definida como GranulMax=[5,5]. O número máximo de regras (MaxReg) no nível III foi definido em função da GranulMax como 25. Neste nível, o número efetivo de regras em cada base de regras pode mudar durante a evolução, pois a mutação e o crossover podem excluir ou reintroduzir regras, gerando uma base de regras mais ou menos compacta, mas que respeite o limite máximo estabelecido.

Os resultados das simulações estão resumidos na tabela 5.1, que traz: o número de ciclos necessários para satisfazer a condição de parada ($\text{Erro}_{tr} \leq 0.01$ ou limite de gerações T definido como 1000); o número efetivo de regras nebulosas que compõem o melhor sistema nebuloso (Regras); o erro de treinamento (Erro_{tr}), com $y_d = f_1(x_1, x_2)$ e considerando-se $N = 225$ e, finalmente, o erro de teste (Erro_{ts}), obtido utilizando-se 2500 padrões de entrada. Com o objetivo de se avaliar a capacidade de generalização do modelo, simulações foram realizadas testando diferentes tipos

de funções no conseqüente das regras TS: função linear (lin) e função não-linear (quad); e dois tipos de base de regras: base de regras específicas (BRE) e genéricas (BRG). Em todos os casos, o método global foi utilizado para otimizar os conseqüentes. A mesma granularidade máxima (GranulMax=(5,5)) e máximo de regras (MaxReg=25) adotados pela abordagem co-evolutiva foram utilizados pelo ANFIS e EGFS. A título de comparação com a abordagem CoevolGFS, o total de indivíduos na população da abordagem EGFS foi definido como 30, representando 30 sistemas nebulosos, e os operadores de crossover e mutação foram aplicados com probabilidade $P_C = 0.2$ e $P_M = 0.08$, respectivamente.

Tabela 5.1: Resultados das simulações: problema de aproximação do mapeamento F_1

Abordagem	Ciclos	Regras	Erro _{tr}	Erro _{ts}
ANFIS	1000	25	0.015	0.016
EGFS	49	23	0.0091	0.033
CoevolGFS-lin-BRE	1000	24	0.013	0.022
CoevolGFS-lin-BRG	1000	25	0.012	0.021
CoevolGFS-quad-BRE	5	23	0.0089	0.013
CoevolGFS-quad-BRG	3	20	0.0093	0.010

A tabela 5.1 mostra que a abordagem CoevolGFS-quad-BRG estabelece a melhor solução, pois o sistema nebuloso resultante desta abordagem alcança (em apenas 3 gerações) melhores resultados em termos de generalização, com um menor número de regras. Na comparação das diferentes abordagens co-evolutivas, o uso de conseqüentes não-lineares (quad) e de bases de regras genéricas (BRG) se mostra mais apropriado para alcançar melhores resultados em termos do erro de teste. As abordagens com funções lineares no conseqüente (incluindo o ANFIS) não atingiram o limite de erro pré-estabelecido ($\text{Erro}_{tr} < 0.01$). Na comparação da evolução com a co-evolução, as vantagens da segunda também ficam evidentes, já que o processo evolutivo tradicional resultou num modelo menos interpretável (mais regras) e com pior capacidade de generalização.

A figura 5.2 ilustra o mapeamento original e a comparação dos resultados de erros de teste, obtidos pelos modelos nebulosos produzidos pelo ANFIS e pelas abordagens EGFS e CoevolGFS-quad-BRE. Os resultados da figura 5.2 comprovam a grande capacidade de aproximação dos modelos resultantes do processo co-evolutivo. Os picos verificados na superfície de erro se devem à ausência de amostras de treinamento justamente naquelas regiões mais elevadas da superfície de aproximação.

A figura 5.3 ilustra a evolução dos erros de treinamento e teste ao longo de 100 ciclos (gerações), caso as condições de parada fossem alteradas para ($\text{Erro}_{tr} = 0$ e $T = 100$). O objetivo aqui é

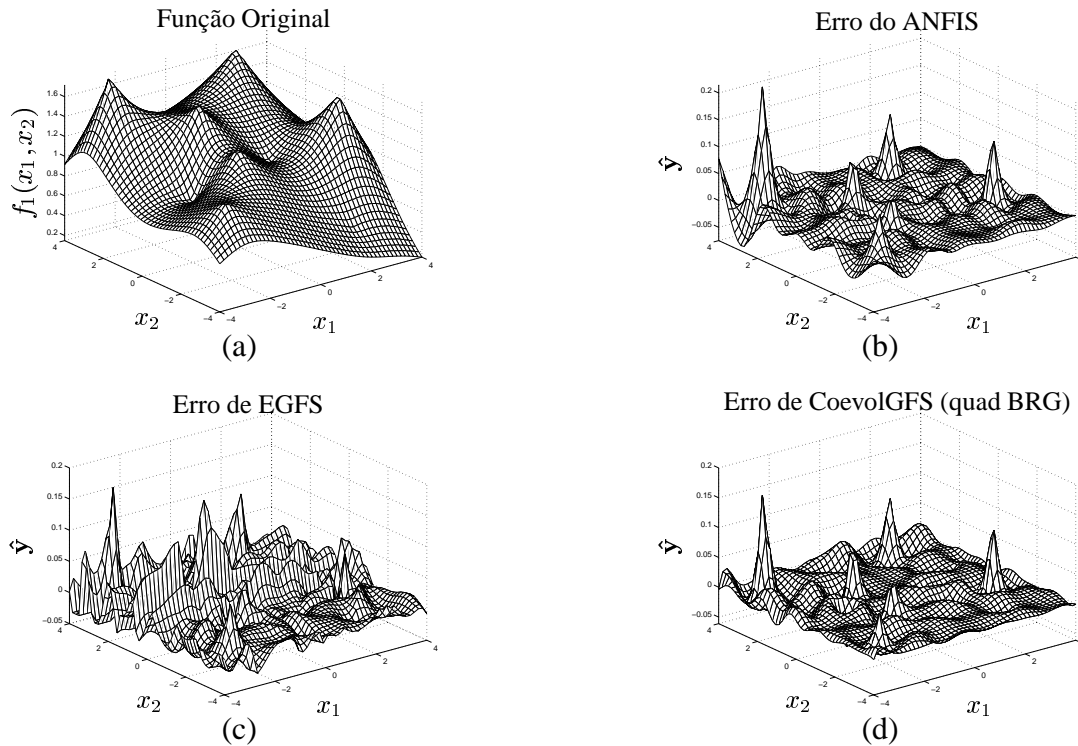


Figura 5.2: Resultados: (a) Função original f_1 ; erro de aproximação produzidos por (b) ANFIS; (c) CoevolGFS-quad-BRE; (d) CoevolGFS-quad-BRG.

testar o comportamento dos sistemas nebulosos que usam somente regras específicas com aqueles que aceitam regras genéricas, ao longo do processo co-evolutivo. Como o método de seleção está associado ao elitismo, o erro de treinamento sempre diminui ou, no mínimo, se mantém inalterado de uma geração para outra. A figura 5.3(b) mostra o valor do erro que seria obtido, se o melhor sistema nebuloso da população a cada geração fosse utilizado para produzir a saída, para todos os 2500 pontos de teste. Nota-se claramente, nas figuras 5.3(a) e 5.3(b), que o uso de bases de regras específicas produz erros de treinamento menores, mas a capacidade de generalização do modelo melhora consideravelmente com a inclusão de regras genéricas. Se os dados de teste fossem utilizados para a validação do modelo de aproximação com BRE, então o processo co-evolutivo deveria ser interrompido antes da geração 17 uma vez que, neste ponto, o erro de teste começa a aumentar. Nas simulações apresentadas na tabela 5.1, como foi utilizado um limite de erro pré estabelecido ($\text{Erro}_{tr} < 0.01$), este foi satisfeito na geração 3 para o caso BRG e na geração 5 para o caso BRE.

A tabela 5.2 resume o custo computacional necessário para se atingir o critério de parada. As abordagens co-evolutivas trabalham com múltiplas populações formadas por centenas de indiví-

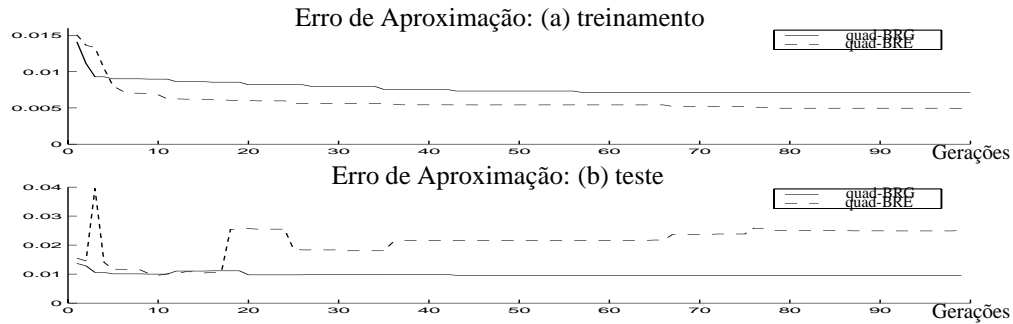


Figura 5.3: Evolução do erro ao longo das gerações.

Tabela 5.2: Comparação de custo computacional

Abordagem	Regras	Ciclos	Tempo de CPU (s)
ANFIS	25	1000	466.26
EGFS	23	49	553.83
CoevolGFS-quad-BRG	20	3	94.74

duos que representam os sistemas nebulosos. Por isso, em geral, o custo computacional destas abordagens para uma geração ($\approx 94.74/3 = 32.58s$), é sempre muito elevado quando comparado com custo do ANFIS para rodar uma época ($\approx 466.26/1000 = 0.47s$). A tabela mostra ainda que a introdução da co-evolução também aumenta o custo computacional, que é menor para a abordagem evolutiva tradicional ($\approx 553.83/49 = 11.3s$). Embora a abordagem co-evolutiva apresente custo computacional por geração superior, nas simulações apresentadas nesta seção, a rapidez dos modelos co-evolutivos em atingir o critério de parada (apenas 3 gerações), garantiu um menor custo no processo de obtenção da solução final. Todavia, conforme será discutido no capítulo 6, a despeito da análise de custo computacional apresentada na tabela 5.2, o dado mais importante diz respeito ao total de regras de cada modelo, já que um modelo menor demandará menos recursos na implementação da solução final.

5.2.2 Aproximação do Mapeamento F_2 : Análise de Interpretabilidade

O problema aqui consiste na aproximação do mapeamento

$$F_2 : \Omega \rightarrow \mathfrak{R}, \text{ onde } f_2(x_1, x_2) = \frac{\sin(x_1)}{x_1} \frac{\sin(x_2)}{x_2},$$

com $\Omega \equiv [-10, 10]^2$, conforme mostrado na figura 5.4(a). A aproximação utiliza N padrões de treinamento (x_{1p}, x_{2p}, f_p) , $p = 1, \dots, N$.

Nesta seção, o enfoque da análise será para o aspecto de interpretabilidade do sistema de aproximação resultante. O problema de aproximação de função será solucionado com o objetivo de comparar os resultados obtidos pelo sistema resultante da co-evolução com aqueles produzidos pelo ANFIS (considerado um sistema parcialmente interpretável), pelo EGFS (cuja interpretabilidade fica comprometida devido à otimização irrestrita das funções de pertinência) e também com os resultados obtidos pelo MLP-BckPr (considerado um modelo do tipo caixa-preta, para o qual não há interpretação lingüística possível). Na abordagem co-evolutiva utilizada nesta seção, os parâmetros ótimos do conseqüente não-linear são estimados pelo método global e o SN utiliza base de regras genéricas.

As abordagens ANFIS e MLP-BckPr foram treinadas durante 6000 épocas (total necessário para que não houvesse grandes variações no erro de treinamento) e os modelos das abordagens CoevolGFS e EGFS evoluíram durante 1000 gerações. O total de indivíduos em cada uma das populações que evoluem é dado por: $S_{IV} = 100$, $S_{III} = 80$, e $S_I = 20$. Os operadores genéticos de crossover e mutação foram aplicados em todos os níveis que evoluem, com probabilidades $P_C = 0.25$ e $P_M = 0.09$, respectivamente. Para a abordagem EGFS, o tamanho da população foi definido como 100. Para as abordagens evolutivas, co-evolutivas e ANFIS, foram testadas duas situações diferentes, no que diz respeito ao objetivo do projeto automático dos sistemas nebulosos:

1. objetivo principal é a interpretabilidade (medida pela compactação da base de regras): neste caso a granularidade máxima foi definida como $\text{GranulMax}=[3,3]$, resultando em $\text{MaxReg}=9$;
2. objetivo principal é a acuidade (medida pelo erro de teste): neste caso a granularidade máxima foi definida como $\text{GranulMax}=[5,5]$, resultando em $\text{MaxReg}=25$.

Os modelos de redes neurais MLP-BckPr foram treinados em uma única situação considerando 25 neurônios na camada intermediária (este total foi escolhido para comparar com o sistema neuro-nebuloso ANFIS que considera 25 regras). Para o ANFIS, diferentes tipos de funções de pertinência foram testados e os menores erros de aproximação foram obtidos com as funções Gaussianas.

A tabela 5.3 traz os resultados dos sistemas nebulosos obtidos por co-evolução (CoevolGFS), por evolução tradicional (EGFS) e dos sistemas de aproximação produzidos pelo ANFIS e MLP-BckPr. Os resultados da tabela 5.3 mostram os erros de treinamento e de teste e o total de neurônios ou regras efetivas dos sistemas (tamanho). O erro de treinamento foi computado para $N = 225$, enquanto que o erro de teste foi encontrado usando-se 2500 padrões.

Tabela 5.3: Resultados das simulações: problema de aproximação do mapeamento F_2

Abordagem	Ciclos	Erro _{tr}	Erro _{ts}	GranulMax	Tamanho
ANFIS	6000	0.042	0.041	[3,3]	9
EGFS	1000	0.015	0.020	[3,3]	9
CoevolGFS	1000	0.010	0.011	[3,3]	8
MLP-BckPr	6000	0.013	0.014	-	25
ANFIS	6000	0.003	0.003	[5,5]	25
EGFS	1000	0.005	0.013	[5,5]	25
CoevolGFS	1000	0.002	0.002	[5,5]	24

A tabela 5.3 mostra o bom desempenho da abordagem CoevolGFS, em termos do compromisso interpretabilidade/acuidade para as duas granularidades testadas. No caso da granularidade menor (GranulMax=[3, 3]), a superioridade da abordagem co-evolutiva (que representa o modelo mais interpretável) fica ainda mais evidente, pois com uma base compacta formada por apenas 8 regras, o sistema nebuloso é mais acurado do que o ANFIS (interpretabilidade média) formado por 9 regras e obteve desempenho comparável às abordagens com interpretabilidade baixa (EGFS com 25 regras e MLP-BckPr com 25 neurônios na camada intermediária - caixa-preta).

A figura 5.4, ilustra a acuidade medida pela capacidade de generalização dos sistemas de aproximação mostrados na tabela 5.3, considerando-se (GranulMax=[3, 3]).

A tabela 5.4 mostra os elementos do sistema nebuloso obtido por co-evolução, para a granularidade [3, 3]. As 8 regras nebulosas, cujo vetor de parâmetros dos consequentes (w_j) representa a função quadrática, utilizam o produto algébrico como operador de agregação de antecedente (A_a). Este parâmetro, que emergiu do processo co-evolutivo, comprova a maior autonomia de projeto, que tem ampla liberdade na escolha da norma mais adequada para agregar os antecedentes na regra, ao contrário do ANFIS que fixa o operador de agregação dos antecedentes como mínimo.

Tabela 5.4: Parâmetros do modelo TS obtido

Regra	A_1^j	A_2^j	w_{j0}	w_{j1}	w_{j2}	w_{j3}	w_{j4}	w_{j5}
1	#	baixo	0.18	0	0.05	0	0	0
2	#	médio	-0.53	0	0.01	0	0.04	0
3	#	alto	0.15	0	-0.04	0	0	0
4	baixo	médio	0	-0.25	-0.02	-0.02	-0.08	0
5	médio	baixo	-5.14	0.04	-1.34	0	-0.08	0.01
6	médio	médio	5.12	-0.09	-0.37	0	-0.15	0.01
7	médio	alto	-4.11	0.03	1.10	0	-0.07	0
8	alto	médio	-0.56	0.38	-0.02	-0.03	-0.08	0

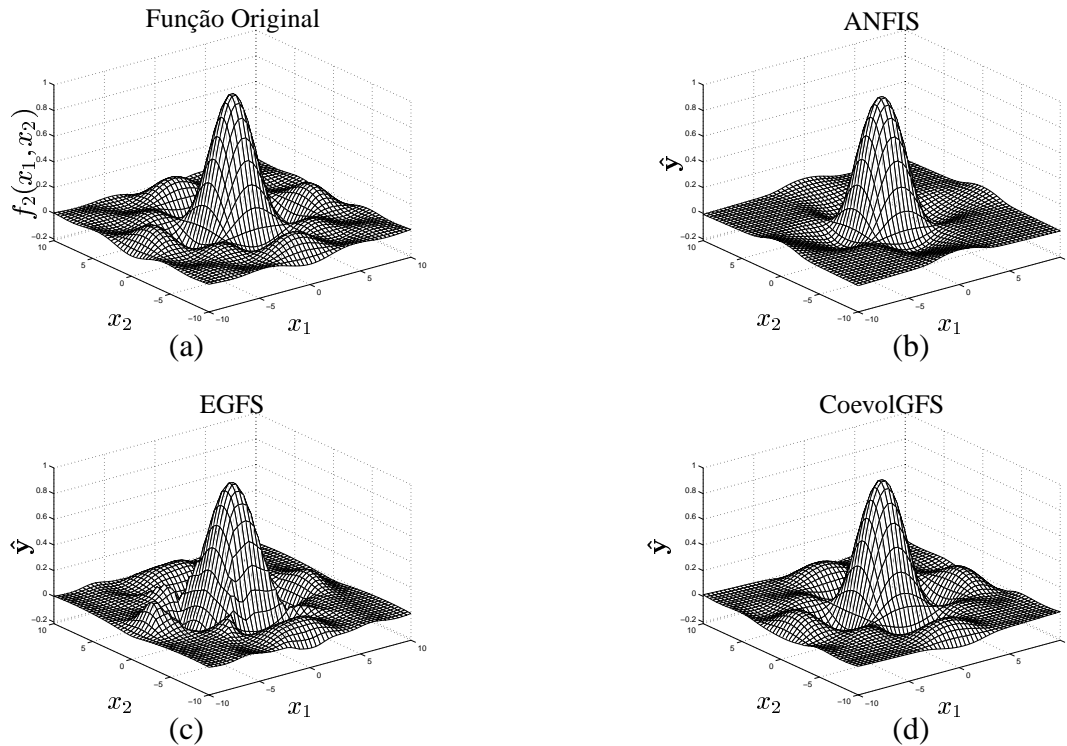


Figura 5.4: (a) Função original f_2 ; aproximação obtida por (b)ANFIS; (c) abordagem evolutiva tradicional; (d) abordagem co-evolutiva.

A tabela 5.4 mostra que o sistema nebuloso resultante atende os critérios de interpretabilidade, pois além de compacta, a base de regras é consistente e formada por regras simplificadas pelo uso de condições irrelevantes, representadas pelo símbolo (#). Conforme discutido na seção 5.1.1, os termos lingüísticos *baixo*, *médio* e *alto* foram definidos *a posteriori*. A base de regras salienta uma característica dos sistemas nebulosos resultantes, ou seja, regras mais gerais atuando juntamente com regras mais específicas, como é o caso, por exemplo, da regra 2 (mais geral) e as regras 4, 6 e 8 (mais específicas). Para que estas regras não atuassem juntas na mesma base de regras, seria necessário que uma delas tivesse mais flexibilidade no conseqüente, permitindo assim a eliminação da outra. Os parâmetros nulos no conseqüente são resultantes da eliminação automática de redundância, patrocinada pelo processo de poda descrito no capítulo 4.

A figura 5.5 traz as partições resultantes do ANFIS após o treinamento e CoevolGFS e EGFS após a evolução. No caso do ANFIS e da abordagem CoevolGFS, o critério de visibilidade é satisfeito, uma vez que os limites máximo e mínimo de sobreposição das funções de pertinência são atendidos. Entretanto, no caso da abordagem EGFS, uma vez que nenhuma restrição foi imposta, o critério de γ -completude para a partição do universo da variável base x_1 não foi atendido.

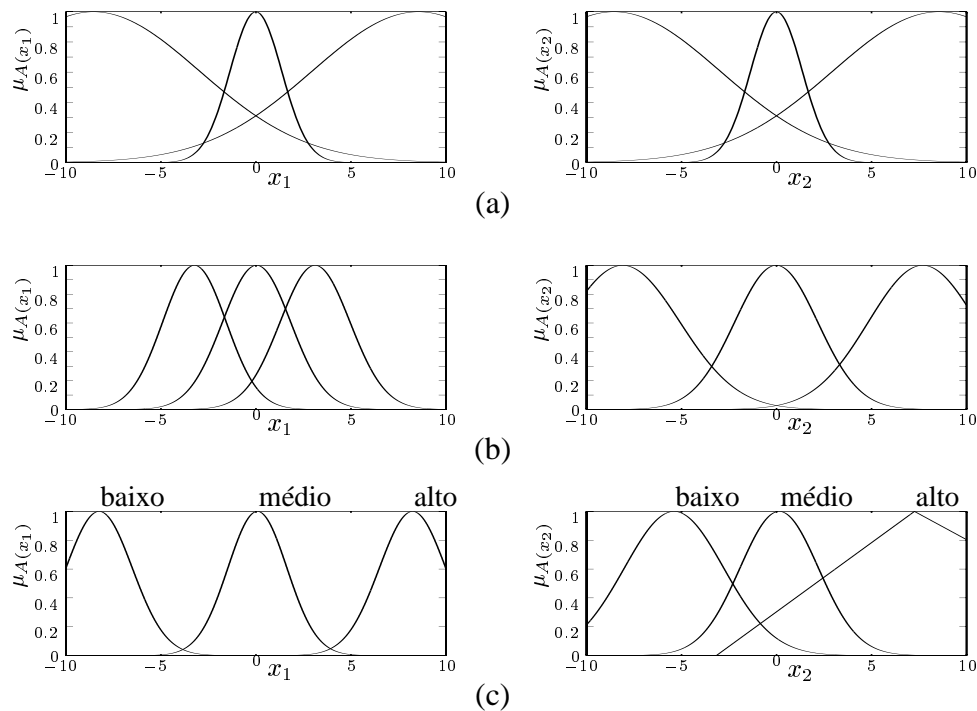


Figura 5.5: Partições nebulosas resultantes (a) do ANFIS; (b) da abordagem EGFS; (c) da abordagem CoevolGFS.

Os resultados desta seção mostram que o aumento da interpretabilidade, resultante de bases de regras consistentes e compactas, com partições nebulosas atendendo o critério de visibilidade, nem sempre acarreta em grande perda de acuidade por parte do modelo. As simulações mostraram que a abordagem CoevolGFS apresentou um bom compromisso entre acuidade e interpretabilidade, com alto grau de autonomia de projeto.

5.3 Aproximação de Funções: Dados com Ruído

O objetivo desta seção é estimar o grau de sobre-ajuste (*overfitting*), resultante da aproximação de funções por modelos nebulosos, na presença de dados com ruído. Isto porque, conforme salientado por Berthold & Hand (1999), modelos extremamente flexíveis, como é o caso dos sistemas nebulosos com conseqüentes não-lineares, se adaptam tão bem que podem incorporar as peculiaridades causadas por perturbações aleatórias. Portanto, nesta seção será analisado o comportamento dos sistemas nebulosos na presença de ruído branco, com distribuição normal. É esperado um bom comportamento, em termos de generalização, para o modelo nebuloso com conseqüentes não-lineares, pois o grande potencial em termos de flexibilidade é controlado automaticamente pelo

processo de poda de parâmetros do conseqüente das regras.

Sempre que dados com ruído são utilizados para construir sistemas de modelagem, o compromisso entre o limiar e a variância do erro de modelagem se torna a maior preocupação, pois estes dados estão diretamente relacionados com a acuidade e capacidade de generalização do modelo (Berthold & Hand, 1999). Com o intuito de analisar este compromisso, será considerado um conjunto Ξ , formado por v sub-conjuntos de padrões de treinamento com ruído, dado por

$$\Xi = \{\xi_1, \dots, \xi_v\},$$

onde cada $\xi_i, i = 1, \dots, v$, representa uma realização independente.

5.3.1 Erro de Previsão

O modelo nebuloso pode ser usado para produzir a saída estimada $\hat{f}(\mathbf{x})$ a partir de um conjunto de treinamento $\xi_i = \xi$. A diferença entre a saída estimada $\hat{f}(\mathbf{x})$ e a realização y da variável aleatória Y é definida como erro de previsão do modelo (Berthold & Hand, 1999). A qualidade de \hat{f} como elemento de previsão de Y , para uma realização específica ξ dos padrões de treinamento, pode ser obtida com base na esperança em relação à distribuição $p(Y|\mathbf{x})$:

$$\text{Esp} \left[\left(Y - \hat{f}(\mathbf{x}|\xi) \right)^2 \right], \quad (5.7)$$

onde $\hat{f}(\mathbf{x}|\xi)$ é a saída estimada pelo modelo para uma realização particular ξ e um determinado \mathbf{x} . Supondo que $\text{Esp}[Y|\mathbf{x}] = f(\mathbf{x})$, a equação 5.7 pode ser reescrita em função de sua parte redutível e irreduzível como:

$$\text{Esp} \left[\left(Y - \hat{f}(\mathbf{x}|\xi) \right)^2 \right] = \underbrace{\left(f(\mathbf{x}) - \hat{f}(\mathbf{x}|\xi) \right)^2}_{\text{parte redutível}} + \underbrace{\text{Esp} \left[(y - f(\mathbf{x}))^2 \right]}_{\text{parte irreduzível}}. \quad (5.8)$$

A segunda parte do erro, denominada de parte irreduzível, mede o erro para a melhor estimacão possível de $\text{Esp}[Y|\mathbf{x}]$. Como esta parte não depende do estimador, o enfoque da análise do erro de previsão será para a primeira parte (redutível).

A parte redutível do erro na equação (5.8) mede a qualidade da estimacão $\hat{f}(\mathbf{x}|\xi)$ para uma realizacão particular ξ . Considerando-se todas as possíveis realizações ($\xi \in \Xi = \{\xi_1 \dots, \xi_\infty\}$), a medida geral da qualidade do estimador \hat{f} pode ser decomposta em suas parcelas de limiar e

variância, tal como se segue:

$$\text{Esp}_\xi \left[\left(f(\mathbf{x}) - \hat{f}(\mathbf{x}|\xi) \right)^2 \right] = \underbrace{\left(f(\mathbf{x}) - \text{Esp}_\xi[\hat{f}(\mathbf{x}|\xi)] \right)^2}_{\text{limiar}} + \text{Esp}_\xi \left[\underbrace{\left(\hat{f}(\mathbf{x}|\xi) - \text{Esp}_\xi[\hat{f}(\mathbf{x}|\xi)] \right)^2}_{\text{variância}} \right] \quad (5.9)$$

Considerando-se que cada realização ξ_i é formada, não por um, mas por um conjunto de padrões distribuídos ao longo do universo, ou seja, $\xi_i = (\mathbf{x}_{pi}, y_{pi})$, $p = 1, \dots, N$, o erro quadrático médio com ruído é definido em função dos seus componentes de limiar e variância, como:

$$\text{EQM}_r(\text{tr}) = \frac{1}{N} \sum_{p=1}^N (\text{Erro}_{\text{limiar}}(\mathbf{x}_p) + \text{Erro}_{\text{var}}(\mathbf{x}_p)), \quad (5.10)$$

onde $\text{Erro}_{\text{limiar}}$ e Erro_{var} correspondem aos componentes de limiar e variância do erro. Nas simulações realizadas, o termo $\text{Esp}_\xi[\cdot]$ vai ser substituído por sua aproximação finita (somatório que considera um número finito de realizações).

5.3.2 Aproximação no \mathfrak{R}^2 Usando Dados com Ruído

Nas simulações realizadas nesta seção, supõe-se que os dados disponíveis para a aproximação da função f_3 são perturbados por um ruído com distribuição normal $\mathcal{N}[\mu, \sigma]$, tal que:

$$F_3 : \Omega \rightarrow \mathfrak{R}, \quad F_3 = f_3 + \mathcal{N}[\mu, \sigma],$$

onde $f_3(x) = \sin(6\pi x) + 2x + \exp(-(1/0.03^2)(x - 0.59)^2)$, $\mu = 0$ e $\sigma = 0.3$, com $\Omega \equiv [0, 1]$, conforme mostrado na figura 5.8(a). Esta função é particularmente interessante porque é suave, mas apresenta uma não-linearidade dominante, dada pelo termo $\sin(6\pi x)$, e outra local, produzida pelo termo $\exp(-(1/0.03^2)(x - 0.59)^2)$.

Dois casos serão analisados: no primeiro caso, um conjunto Ξ com um grande número de realizações para treinamento será utilizado no processo de estimação dos conseqüentes de uma série de modelos TS, com diferentes parâmetros no antecedente: norma de agregação, tipos das funções de pertinência, etc. Neste caso, nenhuma evolução é realizada e o objetivo é apenas testar a estimação dos conseqüentes para diferentes parâmetros no antecedente. Com o objetivo de analisar a influência do ruído no processo evolutivo, um segundo caso será analisado no qual o conjunto Ξ , contendo apenas uma realização $\xi = (\mathbf{x}_p, y_p)$, $p = 1, \dots, N$, será utilizado como conjunto de dados de treinamento no processo co-evolutivo dos sistemas nebulosos.

Caso 1: Influência do ruído na estimação dos conseqüentes TS (análise do compromisso limiar/variância)

Neste caso não há evolução e o objetivo é analisar o compromisso limiar/variância do erro na estimação dos conseqüentes TS, consideraremos 1000 conjuntos de padrões de treinamento ξ_i , cada um formado por $N_i = N = 300$ pares de dados com ruído, resultando em $\Xi_{tr} = \{\xi_1 \cdots, \xi_{1000}\}$. Para cada ξ_i , os parâmetros do conseqüente de diferentes sistemas nebulosos são calculados e a saída $\hat{f}(x)$, para o problema de aproximação de F_3 , é estimada. A tabela 5.5 mostra o erro $EQM_r(tr)$ (definido na equação (5.10)), resultante da estimação usando os dados com ruído, destacando seus componentes de limiar e variância. A média do erro de teste $\overline{EQM}(ts)$ é obtida com base no $EQM(ts)$ resultante de cada estimação, utilizando-se o conjunto $\{T_{s_1}, \dots, T_{s_{1000}}\}$, onde cada T_{s_i} é formado por 3500 dados sem ruído. Nesta etapa de simulações, fixou-se um total de 6 funções de pertinência (GranulMax=6) uniformemente distribuídas no universo de entrada. Como o objetivo é avaliar a influência do ruído no cálculo do conseqüente, a partição nebulosa foi mantida fixa com grau de sobreposição (ov) dado por $ov = \gamma = \kappa = 0.1$.

Tabela 5.5: Componentes de Limiar e Variância do Erro

Estimação do Conseqüente	Limiar	Variância	$EQM_r(tr)$	$\overline{EQM}(ts)$
Global quadrática	0.008	0.005	0.01	0.16
Global constante	0.21	0.002	0.2	2.48
Local quadrática	0.068	0.004	0.07	0.85
Local constante	0.22	0.002	0.2	2.52

A tabela 5.5 mostra as quatro combinações utilizadas para o conseqüente das regras na abordagem co-evolutiva: estimação global dos parâmetros da função quadrática; estimação global dos parâmetros da função constante; estimação local dos parâmetros da função quadrática; estimação local dos parâmetros da função constante. Os resultados mostram que a estimação global conduziu a componentes de limiar extremamente baixos e componentes de variância comparáveis com as demais abordagens. Do ponto de vista de interpretabilidade, como neste experimento as partições são uniformes e mantidas fixas, a diferença se dá nos conseqüentes das regras. Neste caso, a combinação de métodos locais e conseqüentes constantes pode ser mais apropriada para o aspecto de interpretabilidade. Isto se justifica pelas seguintes vantagens resultantes do uso de funções constantes e estimações locais: a transformação dos números reais em termos lingüísticos e a visualização das descrições locais dos parâmetros TS (Jin, 2000; Setnes *et al.*, 1998).

A figura 5.6 ilustra os resultados mostrados na tabela 5.5. Aqui, os componentes de limiar e

variância do erro podem ser visualmente comparados. Os mapeamentos mostrados na figura 5.6 são resultados de partições nebulosas uniformes (alto grau de interpretabilidade). No entanto, nota-se que a função f_3 requer uma maior atenção (mais regras atuando) ao redor do ponto $x = 0.6$, e isto não pode ser definido *a priori*. O fato de que nenhuma das abordagens foi capaz de mapear a não-linearidade local sugere que o ajuste das funções de pertinência (FP) é necessário para se atingir o nível de acuidade desejado.

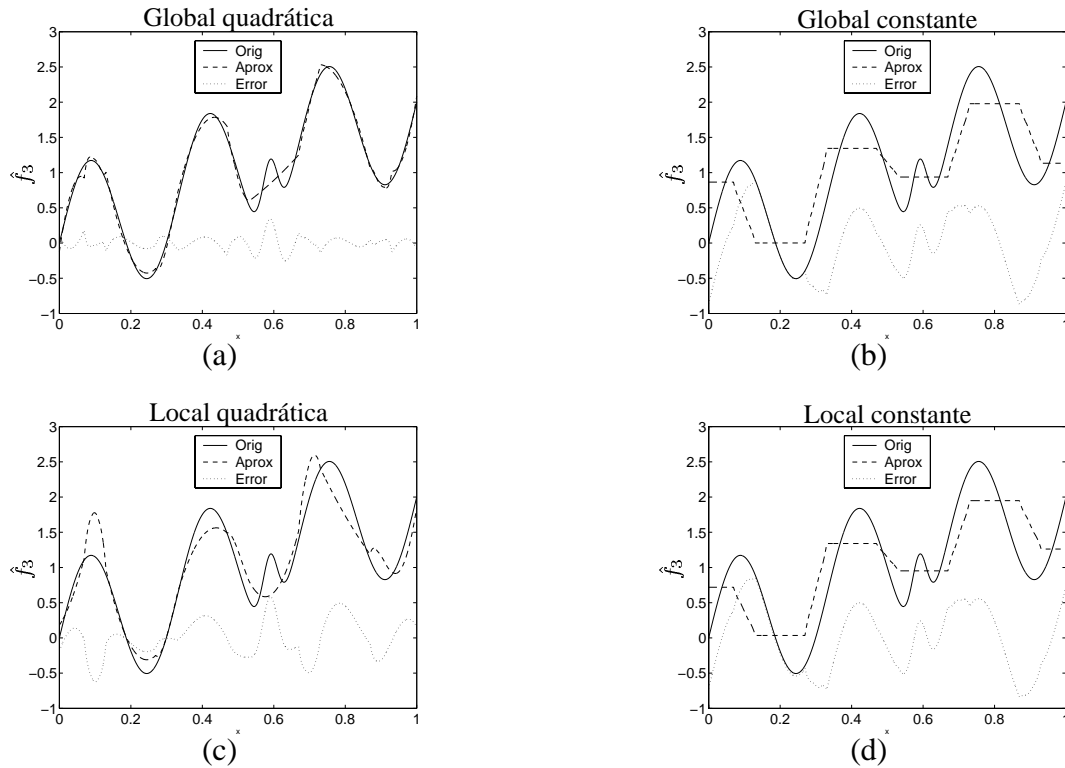


Figura 5.6: Análise gráfica do resultado da otimização dos parâmetros do consequente: (a) otimização global com função não-linear no consequente ; (b) otimização global com função constante no consequente ; (c) otimização local com função não-linear no consequente; (d) otimização local com função constante no consequente.

Uma vez que o ajuste resultante do processo evolutivo altera o formato e a localização (suporte) das funções de pertinência, análises da influência das sobreposições destas funções na estimação dos consequentes são altamente recomendáveis. A figura 5.7 mostra como os componentes de limiar e variância do erro são afetados, considerando-se diferentes graus de sobreposição ($ov = \{0, 0.5, 1\}$) na partição nebulosa da variável x . Nota-se que as abordagens baseadas em estimação local, abreviadas como Lquad e Lct, são mais sensíveis às alterações no grau de sobreposição, tanto para o componente de limiar, quanto para o componente de variância do erro.

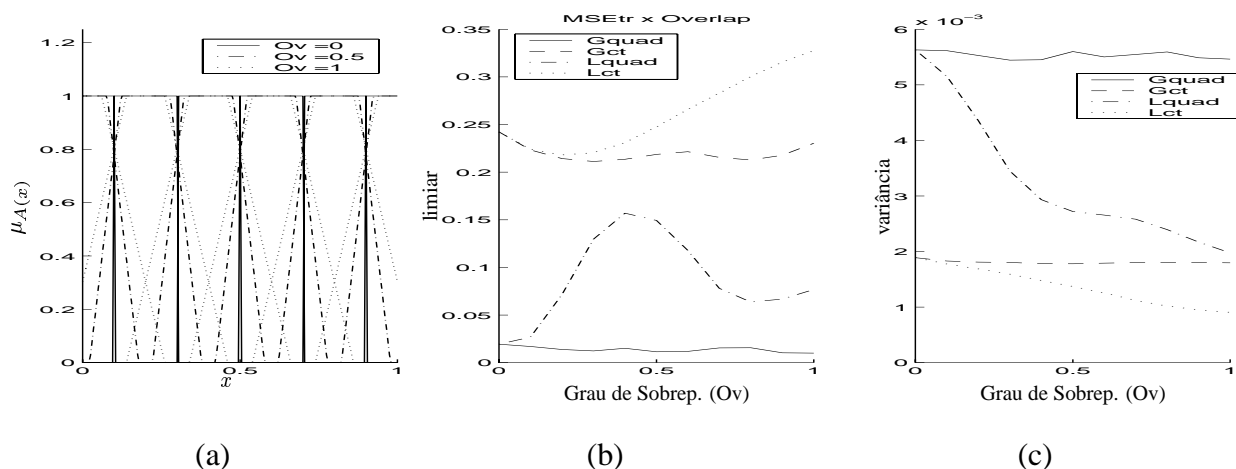


Figura 5.7: Influência da sobreposição das funções de pertinência: (a) variação do grau de sobreposição (as funções de pertinência são trapezoidais); (b) influência sobre o limiar; (c) influência sobre a variância.

Caso 2: Influência do ruído no processo co-evolutivo (análise de sobre-ajuste)

Embora a análise anterior seja importante do ponto de vista teórico, na prática, um número grande de realizações nem sempre é possível. Portanto, para testar as abordagens dentro de um panorama mais próximo da realidade, a co-evolução dos sistemas nebulosos utiliza somente uma realização com ruído, $tr = \xi_i$, na fase de treinamento e um conjunto de teste (ts) sem ruído para avaliar efetivamente o modelo.

A tabela 5.6 mostra o erro de treinamento (EQM(tr)) e teste (EQM(ts)), calculados pela equação (4.2), para os melhores sistemas nebulosos, obtidos dos processos co-evolutivos na geração 100. As mesmas combinações de classe de função e método de otimização testadas no caso 1 são utilizadas nos consequentes das regras. Os resultados dos erros são comparados com o ANFIS treinado com 100 épocas. Em todas as simulações realizadas, supõem-se $N = 300$ padrões de treinamento com ruído $\xi = (x_p, F_{3_p})$, $p = 1, \dots, N$, conforme ilustra a figura 5.8(a), e $N = 3500$ padrões de teste. Como no caso anterior, a GranulMax é definida como GranulMax=6. Uma vez que a aplicação envolve apenas uma variável, este limite também determina o número máximo de regras nebulosas (MaxReg=6) em cada uma das bases de regras.

A figura 5.8 ilustra a capacidade de generalização de algumas das abordagens mostradas na tabela 5.6. Nota-se que a maior flexibilidade associada às abordagens que utilizam estimação global dos parâmetros das funções quadráticas não implica necessariamente em sobre-ajuste, porque qualquer flexibilidade excessiva pode ser eliminada pelo algoritmo de poda apresentado na seção 4.4.4. Com a restrição imposta pelo limite MaxReg=6, a única abordagem capaz de indicar a

Tabela 5.6: Resultados das simulações: problema de aproximação do mapeamento F_3

Abordagem	EQM(tr)	EQM(ts)	Regras
ANFIS	0.0732475	0.103288	6
CoevolGFS Global quadrática	0.070662	0.098229	6
CoevolGFS Global constante	0.139271	0.145969	6
CoevolGFS Local quadrática	0.095581	0.127486	5
CoevolGFS Local constante	0.171676	0.174976	6

não-linearidade local (região em evidência na figura 5.8(c)) foi a que combinou conseqüentes não-lineares com técnicas globais de otimização. Para as demais abordagens, esta particularidade não foi observada porque a forte restrição no número de regras (apenas 6 regras) força o processo de aproximação a focalizar a atenção apenas nas não-linearidades dominantes. Conforme mostram as figuras 5.8(b) e 5.8(d), sistemas nebulosos com conseqüentes lineares (como é o caso do ANFIS) ou constantes (Local constante) necessitam de mais de 6 regras para representar, simultaneamente, as não-linearidades locais e dominantes. Aumentando-se a granularidade até atingir o limite $\text{GranulMax}=\text{MaxReg}=9$, o ANFIS melhorou sua capacidade de aproximação até ser capaz de mapear estas não-linearidades (ver figura 5.8(b) com linhas tracejadas). Este aumento no número de regras, representou um ganho em termos de desempenho para o ANFIS que não foi acompanhado pela abordagem com conseqüente constante. O mapeamento das não-linearidades locais, exigiu da abordagem *CoevolGFS Local constante* um número maior de regras (25 regras para esta aplicação específica).

Os resultados mostram que, para as abordagens com conseqüentes constantes ou lineares, o número de regras necessárias é inversamente proporcional à suavidade da função a ser aproximada, e isso representa uma das grandes desvantagens do uso de sistemas nebulosos com conseqüentes lineares.

A figura 5.9 ilustra a partição nebulosa do melhor indivíduo resultante da abordagem *CoevolGFS Global quadrática* e do sistema nebuloso resultante do ANFIS treinado com 6 regras nebulosas. Conforme mostrado pela figura 5.9(b), mesmo sob fortes restrições, a partição nebulosa da abordagem co-evolutiva, atendeu os critérios de interpretabilidade e o ajuste possibilitou atingir a maior atenção necessária nas redondezas do ponto $x = 0.6$: duas regras atuantes, o que não aconteceu no caso do ANFIS: uma única regra atuante (ver figura 5.9(a)).

A utilização de diferentes classes de funções no conseqüente, juntamente com a possibilidade de escolha entre dois métodos para a obtenção dos parâmetros TS ótimos, produziram modelos com diferentes balanços entre acuidade e interpretabilidade: modelos menos acurados e mais inter-

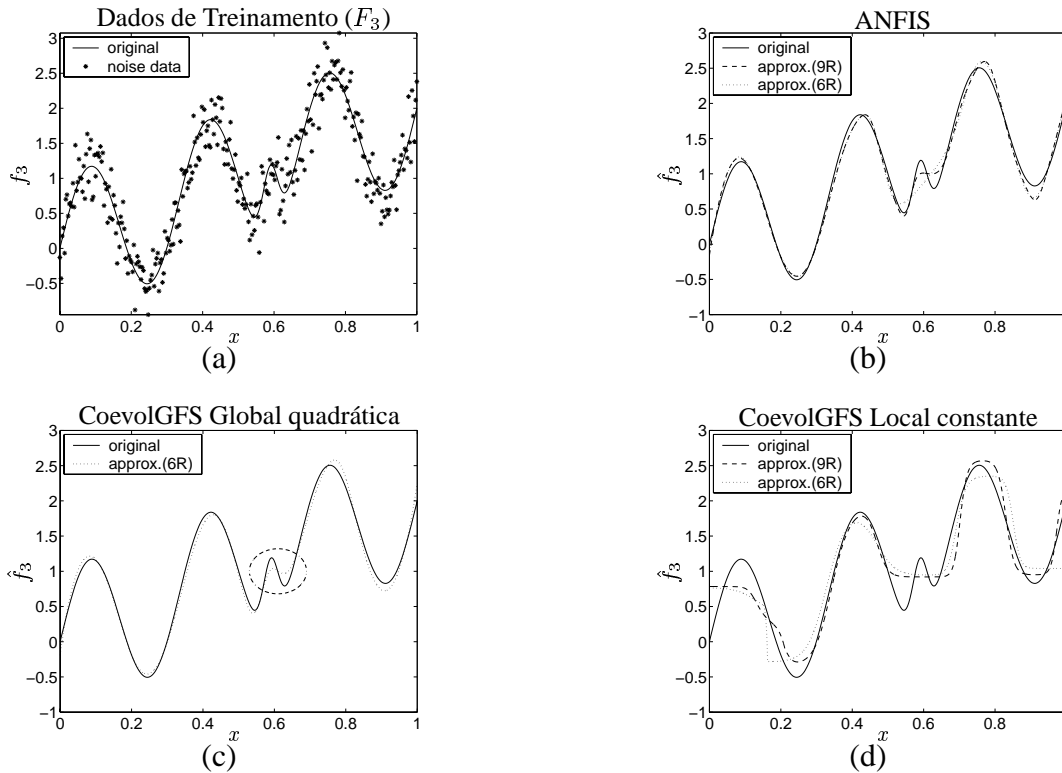


Figura 5.8: Aproximação para dados com ruído (geração 100): (a) Função Original e dados amostrados com ruído; Aproximação dada por: (b) ANFIS que emprega otimização global com função linear no conseqüente (bom desempenho para a região em evidência, quando se empregam 9 regras); (c) CoevolGFS que emprega otimização global com função não-linear no conseqüente (bom desempenho para a região em evidência, com apenas 6 regras); (d) CoevolGFS que emprega otimização local com função constante no conseqüente.

pretáveis, que adotam conseqüentes constantes otimizados por métodos locais; e modelos menos interpretáveis e mais acurados, com conseqüentes não-lineares otimizados por métodos globais.

5.3.3 Aproximação no \mathfrak{R}^3 Usando Dados com Ruído

O problema de aproximação tratado nesta seção avalia a acuidade no espaço tridimensional, considerando dados com ruído. O problema é aproximar a função

$$F_4 : \Omega \rightarrow \mathfrak{R}, \quad F_4(x_1, x_2) = f_4(x_1, x_2) + N[\mu, \sigma],$$

onde $\Omega \equiv [0, 1]^2$, $f_4(x_1, x_2) = 1.9 (1.35 + \exp(x_1) \sin(13(x_1 - 0.6)^2) \exp(-x_2) \sin(7x_2))$, $\mu = 0$ e $\sigma = 0.3$, como ilustrado na figura 5.11(a-b). A função $f_4(x_1, x_2)$ é bem conhecida e aparece em uma série de aplicações apresentadas na literatura (Cherkassky *et al.*, 1996; Rojas *et al.*, 1999).

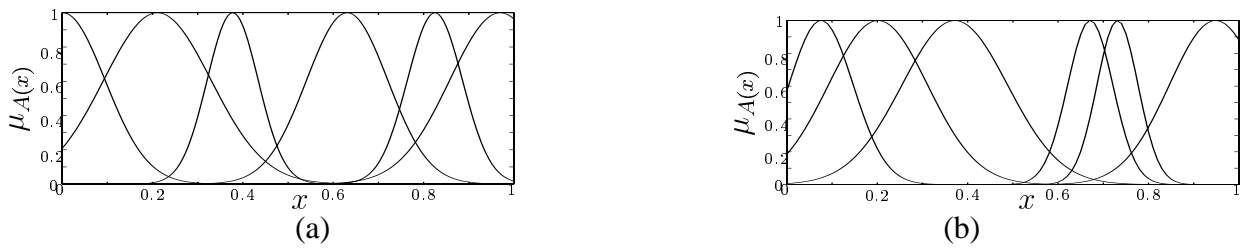


Figura 5.9: A partição nebulosa para o sistema nebuloso obtido por: (a) ANFIS treinado em 100 épocas; (b) abordagem CoevolGFS *Global quadrática* na geração 100.

A aproximação é realizada considerando-se $N = 225$ padrões com ruído (x_{1p}, x_{2p}, F_{1p}) , $p = 1, \dots, N$.

Os modelos evoluíram por 1000 gerações na abordagem co-evolutiva e, como no caso da seção anterior, tem-se: $S_{IV} = 100$, $S_{III} = 80$, e $S_I = 20$. A granularidade máxima das partições nebulosas foi definida como $[3, 3]$. Cada base de regras no nível III contém no máximo 9 regras nebulosas, mas o número efetivo de regras pode ser menor, ao final da co-evolução. As operações de crossover e mutação são realizadas com probabilidades $P_C = 0.3$ e $P_M = 0.1$, respectivamente, para todos os níveis que evoluem (níveis IV, III e I).

A tabela 5.7 traz os resultados para o sistema nebuloso obtido pelo processo co-evolutivo (na geração 1000), assim como a solução produzida pelo ANFIS (treinado com 1000 épocas). Para o ANFIS, a partição do espaço de entrada é fixa e definida como $[3, 3]$. Novamente, diferentes tipos de funções de pertinência foram testados, e os melhores resultados foram obtidos usando-se funções *Gaussianas*. Enquanto os erros de treinamento foram calculados considerando-se $N = 225$ padrões de treinamento com ruído, os testes foram realizados para 2500 pontos.

Tabela 5.7: Resultados das simulações: problema de aproximação do mapeamento F_4

Abordagem	EQM _r (tr)	Erro _{ts}	Regras
ANFIS	0.32	0.21	9
CoevolGFS	0.25	0.13	8

Conforme mostrado na tabela 5.7, a abordagem co-evolutiva (CoevolGFS) atingiu o melhor compromisso entre capacidade de generalização e compactação da base de regras.

A figura 5.10 ilustra a evolução dos erros de treinamento e teste, tanto para o melhor sistema nebuloso da abordagem CoevolGFS (linha cheia), quanto para o ANFIS (linha pontilhada). Como pode se verificar, o erro de treinamento é superior ao erro de teste porque este último é calculado sobre os dados sem ruído.

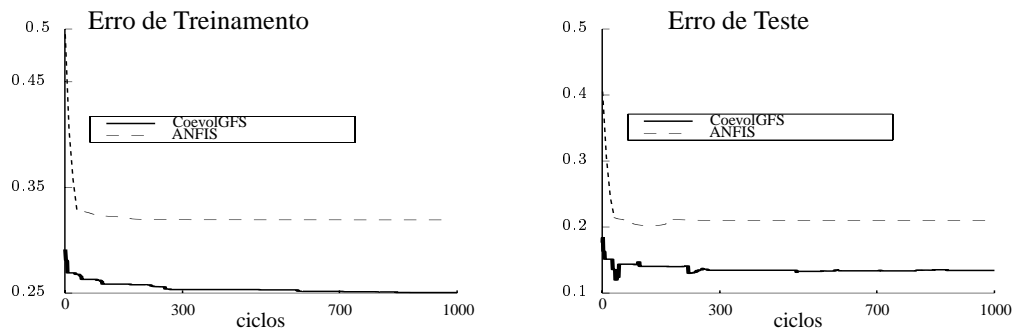


Figura 5.10: Evolução do erro de aproximação de F_4 .

As figuras 5.11(c) e (d) ilustram a capacidade de generalização dos modelos obtidos pelo ANFIS (treinado com 1000 épocas) e pelo processo co-evolutivo ao final de 1000 gerações.

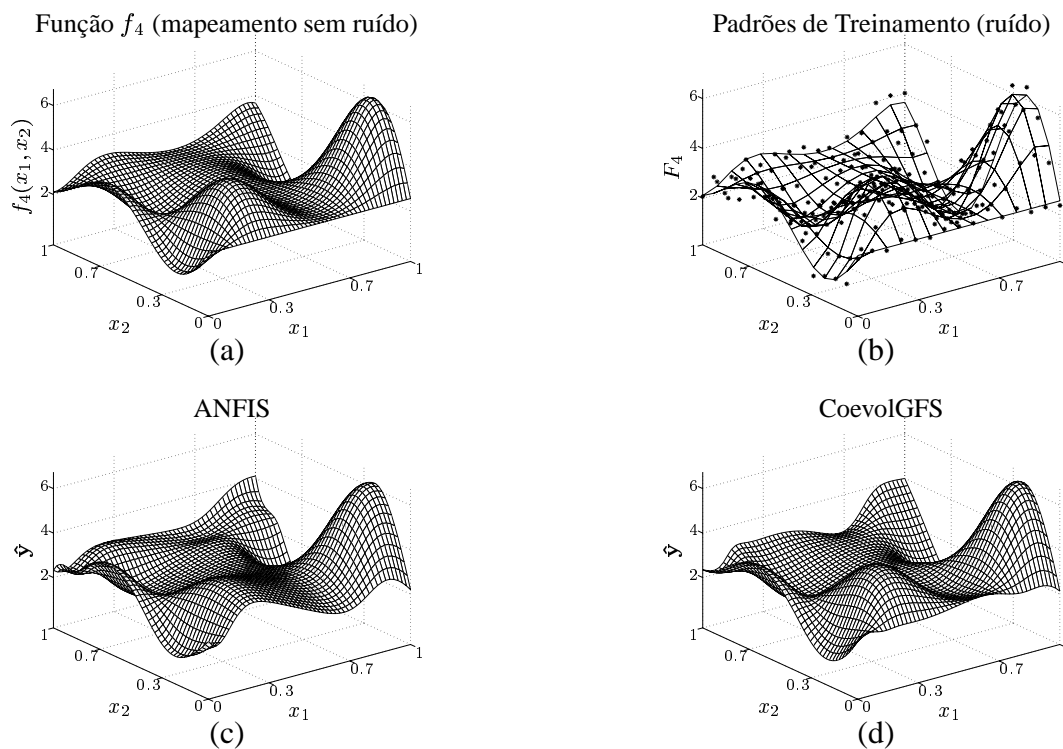


Figura 5.11: (a) Função original $f_4(x_1, x_2)$; (b) Dados de treinamento; Aproximação dada pelo ANFIS; (d) Aproximação dada por CoevolGFS.

Nota-se que, mesmo sob o aumento da flexibilidade resultante do uso de conseqüentes não-lineares, o processo de poda do algoritmo proposto na seção 4.4.4 elimina as redundâncias e reduz o sobre-ajuste. E mais, as restrições impostas pelo critério da visibilidade produzem partições que atendem o critério de γ -completude e κ -sobreposição, conforme mostra a figura 5.12(b). No caso do ANFIS (figura 5.12(a)), os critérios de visibilidade também foram atendidos, no entanto a

produção de partições do universo com maior granularidade resultou em uma base de regras maior, reduzindo um pouco a interpretabilidade do modelo.

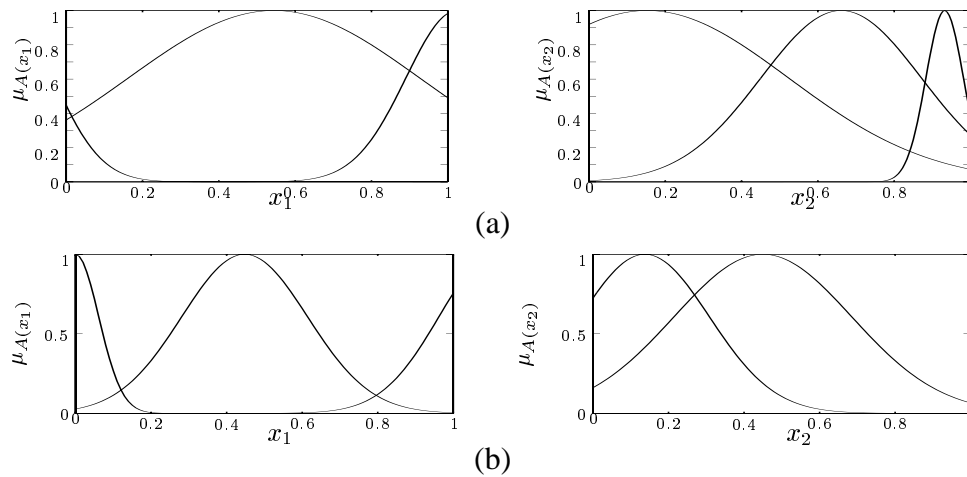


Figura 5.12: Partições nebulosas resultantes (a) do ANFIS; (b) da abordagem CoevolGFS.

5.4 Problemas de Classificação

Nesta seção, o objetivo é avaliar o desempenho dos sistemas nebulosos num contexto de classificação de padrões, resultando em classificadores baseados em regras nebulosas, ou simplesmente, classificadores nebulosos. Numa primeira etapa, os dados das espirais concêntricas (Lang & Witbrock, 1988) serão utilizados para avaliar a eficiência dos modelos obtidos para solucionar um problema de classificação complexo, envolvendo duas variáveis. Numa segunda etapa, o desempenho do modelo co-evolutivo, em espaço de dimensão mais elevada, é investigado através da utilização dos dados de classificação da IRIS. Em ambos os casos, os resultados do modelo proposto são comparados aos resultados obtidos com o ANFIS, sendo que no caso da IRIS, o modelo nebuloso obtido por evolução tradicional e o modelo proposto por Castellano & Fanelli (2000) também serão avaliados.

5.4.1 Dados das Espirais Concêntricas

O problema das espirais concêntricas é bastante conhecido na literatura e teve origem no campo das redes neurais artificiais. De uma forma geral, as propostas co-evolutivas têm obtido bons resultados. Por exemplo, Juillé & Pollack (1996) solucionaram o problema das espirais, co-evoluindo uma

estrutura de programação genética, e Potter & De Jong (2000) co-evoluíram uma rede neural com 12 neurônios escondidos, capaz de classificar corretamente todos os padrões de treinamento. O objetivo aqui será solucionar este problema através da co-evolução de parâmetros de um sistema nebuloso, produzindo um modelo interpretável (mais compacto).

Supõe-se que os dados de entrada são as coordenadas $\mathbf{x}_p = (x_1, x_2)$; $p = 1, \dots, N$; das duas espirais que têm origem no ponto $(0, 0)$, conforme mostrado na figura 5.13.

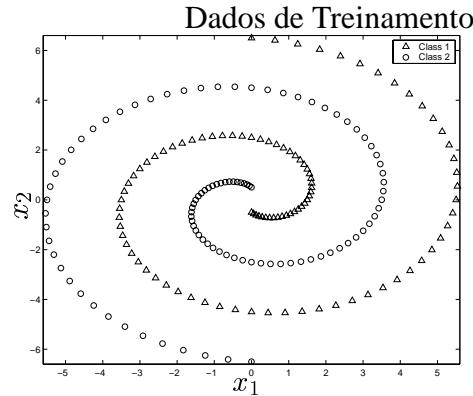


Figura 5.13: Dados de treinamento no problema das espirais.

Para desenvolver o classificador nebuloso supõe-se que, se $f(\mathbf{x}_p)$ é a saída do sistema nebuloso para o padrão de entrada \mathbf{x}_p , então a seguinte atribuição é realizada para relacionar o valor de $f(\mathbf{x}_p)$ com a classe Cl_i :

$$\text{Classe} = \begin{cases} Cl_1 & \text{se } \text{round}(f(\mathbf{x}_p)) = +1 \\ Cl_2 & \text{se } \text{round}(f(\mathbf{x}_p)) = -1 \end{cases} \quad (5.11)$$

onde,

$$\text{round}(a) = \begin{cases} \text{int}(a) - 1 & \text{se } a - \text{int}(a) \leq -0.5 \\ \text{int}(a) & \text{se } -0.5 < a - \text{int}(a) < 0.5 \\ \text{int}(a) + 1 & \text{se } a - \text{int}(a) \geq 0.5 \end{cases}$$

Aqui, o principal objetivo é o desenvolvimento de um classificador nebuloso, capaz de associar cada coordenada (x_1, x_2) com a espiral apropriada. As coordenadas são rotuladas para indicar a qual das duas espirais elas se referem. Um padrão de treinamento é representado por uma tripla (x_1, x_2, Cl) , onde $Cl = +1$ para uma espiral, e $Cl = -1$ para a outra. Dois conjuntos são gerados e cada um é composto por 97 pontos. Os classificadores nebulosos são co-evoluídos usando-se o conjunto formado por $N = 194$ pontos, 97 para cada espiral.

Isto significa que: se a saída $f(\mathbf{x}_p) \in [0.5, 1.5)$ então a classe é Classe 1 (representada pela região em branco na figura 5.15); se a saída é $f(\mathbf{x}_p) \in [-0.5, -1.5)$ a classe é Classe 2 (região em cinza na figura 5.15). Se a saída não pertence a nenhum destes intervalos a classe é indeterminada (região em preto na figura 5.15).

O tamanho das populações para cada espécie é dado por: $S_{IV} = 100$, $S_{III} = 80$, e $S_I = 20$. As operações de crossover e mutação são realizadas com probabilidades $P_C = 0.3$ e $P_M = 0.1$, respectivamente, para os três níveis que evoluem. As partições nebulosas adotam uma granularidade máxima dada por $[3,3]$, com as funções de pertinência uniformemente distribuídas no intervalo $[-6, 6]$, para a variável x_1 , e $[-6.5, 6.5]$, para a variável x_2 . Com o objetivo de produzir uma base de regras extremamente compacta, o número máximo de regras foi definido como $\text{MaxReg} = 4$.

A figura 5.14 mostra o total de classificações erradas para o melhor indivíduo (linha cheia) e para a média da população (linha pontilhada), ao longo do processo co-evolutivo. O melhor indivíduo na primeira geração classifica 45 pontos de forma incorreta, mas após 529 gerações o classificador nebuloso resultante é capaz de classificar corretamente todos os pontos.

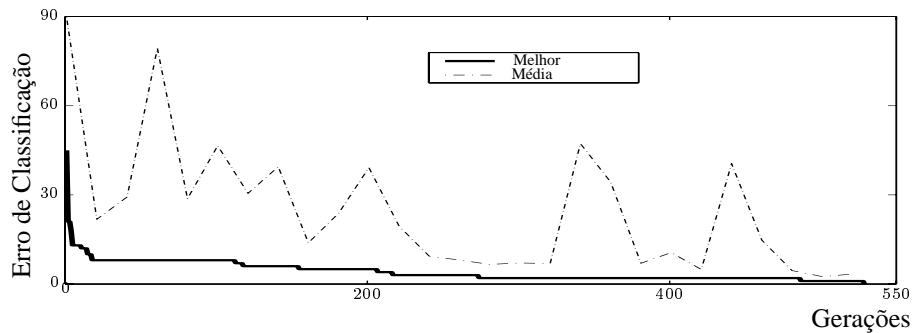


Figura 5.14: Erro de classificação ao longo do processo co-evolutivo.

A tabela 5.8 resume o desempenho do classificador nebuloso resultante do processo co-evolutivo (num total de 1000 gerações), comparado com aquele resultante do ANFIS (treinado com 1000 épocas). O ANFIS opera com uma granularidade fixa, definida como $[3, 3]$, e funções de pertinência *Gaussianas*. As mesmas atribuições de classes usadas pelo sistema co-evolutivo são adotadas pelo ANFIS, para relacionar a saída real a uma das duas classes possíveis.

A figura 5.15 ilustra o desempenho dos classificadores nebulosos resultantes da abordagem co-evolutiva e do ANFIS, mostrando que o mapeamento produzido pela abordagem co-evolutiva está mais próximo dos dados originais.

Tabela 5.8: Resultados das simulações para o problema das espirais.

Abordagem	Ciclos	Regras	Erro de Classificação
ANFIS	1000	9	18
CoevoIGFS	529	4	0

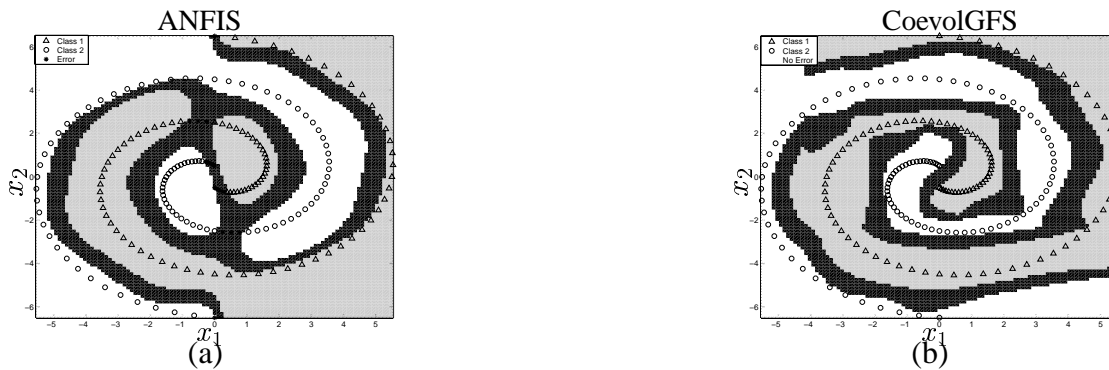


Figura 5.15: Desempenho dos classificadores nebulosos resultantes (a) do ANFIS; (b) da abordagem CoevolGFS.

Verifica-se que a classificação produzida pelo ANFIS, foi considerada insatisfatória (18 erros). Testes realizados com outros tipos de funções de pertinência (trapezoidal, triangular) e ampliação do máximo de gerações ($T = 6000$) não trouxeram melhora no desempenho do ANFIS. Isto mostra que, para o total de regras estipulado (9 regras definidas pela granularidade $[3, 3]$), o mapeamento determinado pelos consequentes lineares do ANFIS não é capaz de separar corretamente as classes. Somente com o aumento da granularidade para $[4, 4]$ (num total de 16 regras), o sistema foi capaz de classificar corretamente todos os 194 padrões.

As regras que compõem a BR do classificador nebuloso, obtido por co-evolução na geração 529, são mostradas a seguir:

$$R_1 : \text{Se } x_1 \text{ é baixo E } x_2 \text{ é baixo então } y = -0.31 + 1.6x_1 - 0.26x_2 + 0.34x_1^2 + 0.17x_2^2 - 0.1x_1x_2$$

$$R_2 : \text{Se } x_1 \text{ é médio E } x_2 \text{ é baixo então } y = 15.27 - 1.29x_1 + 7.76x_2 - 0.05x_1^2 + 0.84x_2^2 - 0.46x_1x_2$$

$$R_3 : \text{Se } x_1 \text{ é médio E } x_2 \text{ é alto então } y = -17.21 - 2.21x_1 + 7.60x_2 - 0.08x_1^2 - 0.78x_2^2 + 0.45x_1x_2$$

$$R_4 : \text{Se } x_1 \text{ é alto E } x_2 \text{ é alto então } y = 1.14 + 2.07x_1 + 1.24x_2 - 0.25x_1^2 - 0.28x_2^2 - 0.34x_1x_2$$

O operador de agregação dos antecedentes representando a conjunção (E) obtido pelo processo co-evolutivo foi a norma triangular t_5 , com parâmetro associado $p_t = 9.96$. Conforme pode se

verificar, a base de regras é extremamente compacta e consistente. As funções de pertinência associadas aos termos lingüísticos *baixo*, *médio* e *alto* estão ilustradas na figura 5.16. A partição nebulosa, atende os critérios de visibilidade.

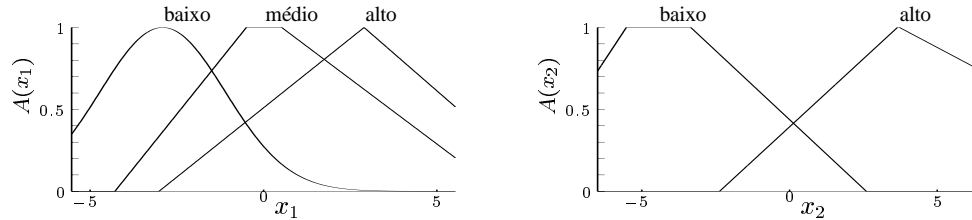


Figura 5.16: Termos lingüísticos associados às funções de pertinência para as variáveis de entrada x_1 e x_2 .

Notadamente, a abordagem co-evolutiva resultou num sistema nebuloso classificador extremamente compacto, atendendo os demais critérios de interpretabilidade, e cujo desempenho atingiu os níveis de acuidade desejados, sem a necessidade da intervenção do projetista para definir parâmetros cruciais, como as funções de pertinência, a base de regras e o operador de agregação dos antecedentes das regras.

5.4.2 Dados da IRIS

Os classificadores nebulosos resultantes da co-evolução serão agora avaliados em um outro contexto, considerando-se os padrões de entrada conhecidos como dados de classificação da IRIS. Popularizado por Fisher (1936), este problema de classificação, envolvendo três classes, possui 150 vetores pertencentes ao \mathbb{R}^4 , representando 50 exemplos de cada espécie de flores: *Iris setosa* (Cl_1), *Iris versicolor* (Cl_2), e *Iris virginica* (Cl_3). Os padrões de treinamento $x_p = (x_{1p}, x_{2p}, x_{3p}, x_{4p})$, $p = 1, \dots, 150$, foram normalizados dentro do intervalo $[0, 1]$. Os atributos são os seguintes: x_{1p} mede o comprimento da sépala da flor, x_{2p} mede a largura da sépala, x_{3p} mede o comprimento da pétala e x_{4p} mede a largura da pétala.

Os processos de treinamento e teste dividem o conjunto de dados disponíveis, utilizando 75 padrões cada um (25 padrões escolhidos aleatoriamente para cada classe). Os propósitos desta divisão são: avaliar a capacidade de generalização do modelo no contexto de classificação e comparar com os resultados obtidos por Castellano & Fanelli (2000).

Tomando-se $f(x_p)$ como a saída do classificador nebuloso (para um padrão de entrada x_p), obtido pelo processo co-evolutivo, a seguinte atribuição é realizada para relacionar a saída $f(x_p)$

com cada uma das classes Cl_i :

$$Cl_1 \rightarrow \text{round}(f(\mathbf{x}_p)) = 1 ,$$

$$Cl_2 \rightarrow \text{round}(f(\mathbf{x}_p)) = 2 ,$$

$$Cl_3 \rightarrow \text{round}(f(\mathbf{x}_p)) = 3 ,$$

onde $\text{round}(\cdot)$ é dado pela equação (5.11).

Os tamanhos das populações de cada espécie do processo co-evolutivo são: $S_{IV} = 30$, $S_{III} = 70$ e $S_I = 10$. As operações de crossover e mutação são realizadas com probabilidades $P_C = 0.2$ e $P_M = 0.08$, respectivamente, para os três níveis que evoluem. A abordagem CoevolGFS supõe BRG e otimização global dos parâmetros da função não-linear nos conseqüentes TS. Com o objetivo de comparar com os resultados obtidos por Castellano & Fanelli (2000), o número máximo de regras em cada base de regras foi definido como MaxReg=4. A granularidade máxima é definida como GranulMax=[2, 2, 2, 2]. Para a abordagem EGFS, adota-se uma população composta por 30 indivíduos, com as mesmas taxas de operadores genéticos adotadas pela abordagem CoevolGFS. O ANFIS e a abordagem EGFS operam com a mesma granularidade do modelo co-evolutivo (GranulMax=[2, 2, 2, 2]), resultando num máximo de regras dado por MaxReg=16, para as duas abordagens.

A tabela 5.9 mostra os erros de classificação de treinamento e teste obtidos pela abordagem de Castellano & Fanelli (2000), ANFIS, EGFS e CoevolGFS.

Tabela 5.9: Resultados das simulações para o problema da IRIS

Abordagem	Ciclos	Regras	Erros de Classificação	
			Treinamento	Teste
CompacFuzzClass Castellano & Fanelli (2000)	-	5	0	4
ANFIS	100	16	0	6
EGFS	100	13	0	8
CoevolGFS	100	4	0	3

Conforme mostrado na tabela 5.9, o classificador nebuloso resultante da co-evolução apresenta um desempenho superior às demais abordagens, uma vez que este obtém o melhor compromisso entre acuidade e interpretabilidade (medida pela compactação da base de regras).

A base de regras produzida pela abordagem CoevolGFS é consistente e apresenta condições irrelevantes em todas as regras, tornando-as mais simples (mais curtas), conforme mostrado a seguir:

$$R_1 : \text{Se } x_3 \text{ é baixo E } x_4 \text{ é baixo então } y = 0.86 - 0.3x_1 + 0.19x_2 + 0.31x_3 + 0.09x_4 - 0.14x_1^2 - 0.23x_2^2 - 2.86x_3^2$$

R_2 : Se x_1 é baixo E x_2 é baixo E x_3 é alto então $y = 1.15 - 1.97x_1 - 6.06x_2 + 1.82x_3 + 3.41x_4 + 5.18x_2^2$

R_3 : Se x_1 é alto E x_3 é baixo E x_4 é alto então $y = 9.7 - 0.17x_1 + 1.97x_2 + 2.65x_3 - 33.7x_4 + 0.36x_1^2 - 3.44x_2^2 - 0.8x_3^2 + 31x_4^2$

R_4 : Se x_1 é alto E x_2 é alto E x_4 é baixo então $y = 1.52 - 0.49x_1 + 0.2x_2 + 3.83x_3 + 0.32x_4 + 0.29x_1^2 - 0.42x_2^2 - 2.42x_3^2 - 0.24x_4^2 + 0.33x_1x_2x_3x_4$

O operador de agregação do antecedente representando a conjunção (E) é dado pela t-norma t_1 , mostrada no capítulo 3, seção 3.1.3, com parâmetro associado $p_t = 1.66$. As funções de pertinência associadas aos termos lingüísticos *baixo* e *alto* são mostradas na figura 5.17. Novamente, nota-se

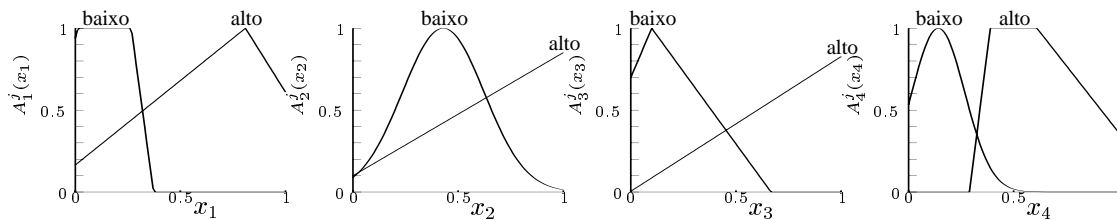


Figura 5.17: Partição nebulosas para as variáveis x_1 , x_2 , x_3 , e x_4 .

um bom nível de acuidade, obtido com uma base de regras compacta, e uma partição nebulosa que atende os critérios de interpretabilidade.

É importante salientar que existem muitos resultados na literatura que conseguiram 100% de classificações corretas quando, no processo de treinamento, são escolhidos os 150 padrões disponíveis, i.e. dados de treinamento ($Tr = 150$) e dados de teste ($Ts = 0$). Treinada com a totalidade dos padrões disponíveis, a abordagem CoevolGFS também foi capaz de classificar corretamente todos os dados. No entanto, como objetivo aqui é avaliar a capacidade de generalização do modelo, toda a ênfase foi dada para as simulações com dados de treinamento ($Tr = 75$) e dados de teste ($Ts = 75$).

A tabela 5.10 ilustra os desempenho em termos de custo computacional, requerido pelo modelo co-evolutivo e pelo ANFIS, quando contabiliza-se o mesmo número de ciclos para cada abordagem. Conforme já salientado na seção 5.2.1, os menores valores no caso do ANFIS podem ser justificados pelo fato de 1 ciclo (época) do sistema ANFIS requerer um passo incremental de otimização dos parâmetros do antecedente das regras e do conseqüente para apenas um sistema nebuloso. Para as abordagens evolutiva e co-evolutiva, 1 ciclo (geração) envolve o ajuste do antecedente e otimização do conseqüente, não para um, mas para uma população de sistemas nebulosos. O maior custo

computacional da abordagem co-evolutiva, justifica-se pela obtenção de um sistema mais parcimonioso, com um menor número de regras (apenas 4). Este aspecto será discutido mais profundamente na conclusão deste trabalho (capítulo 6).

Tabela 5.10: Comparação de custo computacional

abordagem	Regras	Ciclos	Tempo de CPU(s)
ANFIS	16	100	16
EGFS	13	100	1074
CoevoIGFS	4	100	41290

5.5 Experimentos de Caminho Fechado

Toda vez que um modelo nebuloso com alto grau de flexibilidade é proposto, uma questão surge: qual seria o desempenho do modelo se os dados de treinamento fossem gerados por um sistema nebuloso simples? Em outras palavras, em que nível pode ser medida a capacidade de projeto do sistema que está sendo proposto, para reproduzir as particularidades de um elemento específico mais simples no espaço de busca, escolhido arbitrariamente para gerar os dados de treinamento? Este processo, denominado experimento de caminho fechado (Iyoda & Von Zuben, 2002), será utilizado para testar, sob diferentes níveis de dificuldade, a capacidade de busca do modelo co-evolutivo, proposto para o projeto automático de sistemas nebulosos.

Suponha que os dados de treinamento sejam gerados pelo sistema nebuloso exemplificado na seção 4.4.1, cujos parâmetros serão reproduzidos na tabela 5.11 para maior comodidade do leitor. Este sistema, denominado Sistema Nebuloso Original (SNO), é composto por 5 regras nebulosas $R_j, j = 1, \dots, 5$, envolvendo duas variáveis $x_k, k = 1, 2$, da forma:

$$R_j : \text{ Se } X_1 \text{ é } A_1^j \text{ E } X_2 \text{ é } A_2^j \text{ então } Y = w_{j0} + w_{j1}x_1 + w_{j2}x_2 + w_{j3}x_1^2 + w_{j4}x_2^2 + w_{j5}x_1x_2 .$$

onde a conjunção (E) é dada pelo **min**. A título de comparação, a base de regras e a partição nebulosa que formam o SNO serão novamente mostradas na tabela 5.11 e na figura 5.18(a), respectivamente. O mapeamento resultante do processo de inferência no (SNO), será também ilustrado na figura 5.19(a).

Com o objetivo de mostrar a capacidade de busca do sistema proposto, foram definidos três níveis de dificuldade, variando de um nível de dificuldade menor (nível A), passando por um nível intermediário (nível B), até um nível de dificuldade maior (nível C), conforme detalhado a seguir:

- nível A: A granularidade máxima é definida como $\text{GranulMax}=[3, 3]$ e o número máximo de regras é definido como $\text{MaxReg}=5$. Como estes valores são os mesmos valores do sistema nebuloso original, a busca fica direcionada para a composição das regras (combinação dos índices que identificam as FP) e a partição nebulosa (formato e localização das FP). O método de busca parte de uma partição nebulosa uniforme na geração inicial, para tentar atingir a partição nebulosa desejada (mostrada na figura 5.18(a)), tentando sempre diferentes base de regras (todas formadas por 5 regras individuais). Não existe a possibilidade de exclusão de regras na base de regras;
- nível B: A granularidade máxima é definida como $\text{GranulMax}=[3, 3]$ e o número máximo de regras é ampliado para $\text{MaxReg}=9$. Neste caso, somente a granularidade máxima (GranulMax) é igual à do sistema original. Todos os demais parâmetros do sistema nebuloso original, incluindo o número de regras efetivas, a combinação dos termos lingüísticos, e a partição nebulosa, devem ser encontrados pela abordagem co-evolutiva. A exclusão de regras na base de regras é possível.
- nível C: A granularidade máxima é ampliada para $\text{GranulMax}=[5, 5]$ e o número máximo de regras é definido como $\text{MaxReg}=25$. Este é o caso mais difícil, pois a abordagem co-evolutiva deve ser capaz de eliminar um grande número de regras desnecessárias, encontrar a combinação ótima na base de regras e, ainda, eliminar termos redundantes e ajustar os demais termos da partição nebulosa. A exclusão de regras é possível.

Independente do nível de dificuldade, o problema do caminho fechado consiste em se encontrar o modelo TS apropriado, baseado apenas nos dados de treinamento obtidos pelo mapeamento produzido pelo SNO, ou seja, geram-se dados de treinamento de um sistema nebuloso específico e a partir destes dados busca-se reproduzir o próprio sistema nebuloso que gerou os dados. Aqui, o conjunto de treinamento é composto por 225 padrões. A abordagem co-evolutiva com BRG e otimização global dos consequentes não-lineares, submetida aos três níveis (A, B e C) de dificuldade na busca, é utilizada para se encontrar o conjunto de parâmetros mais apropriados para tentar reproduzir o mapeamento original a partir dos dados disponíveis. Para os níveis B e C, um fator de penalização foi incluído no cálculo do *fitness* para privilegiar os sistemas nebulosos com bases mais compactas. No processo co-evolutivo o total de indivíduos em cada uma das populações é: $S_{IV} = 100$, $S_{III} = 50$, e $S_I = 40$. As operações de crossover e mutação são realizadas com probabilidade de $P_C = 0.3$ e $P_M = 0.1$, respectivamente, para todos os níveis que evoluem.

A tabela 5.11 traz os parâmetros do sistema nebuloso original (SNO) e os parâmetros dos sistemas nebulosos resultantes do processo co-evolutivo, para os três níveis de dificuldade.

Tabela 5.11: Os parâmetros dos sistemas nebulosos resultantes

Nível	Regra	A_1^j	A_2^j	w_{j0}	w_{j1}	w_{j2}	w_{j3}	w_{j4}	w_{j5}
SNO	1	baixo	baixo	-1	0	0	0	0	0
	2	baixo	alto	1	0	0	-0.19	0.47	0
	3	médio	médio	-0.43	-0.37	-0.35	0.62	-0.77	0.5
	4	alto	baixo	2	-0.5	0	0	0	0.85
	5	alto	alto	1	0.13	0.22	0	0	0
A	1	baixo	baixo	-0.97	-0.95	0.01	2.71	-0.59	3.70
	2	baixo	alto	13.27	0.96	-30.31	-13.11	19.47	0.62
	3	médio	médio	-1.19	2.18	0.08	1.17	-0.12	-2.95
	4	alto	baixo	1.70	0.58	-3.26	-0.90	-9.49	6.75
	5	alto	médio	-10.06	31.68	-13.77	-12.66	16.60	-8.29
B	1	baixo	baixo	-0.97	-1.27	-0.25	6.33	0.36	0.22
	2	baixo	alto	19.16	-42.62	-33.39	-28.83	16.17	47.59
	3	médio	médio	0.38	-4.04	-0.52	6.14	-0.43	-0.62
	4	alto	baixo	2.05	-0.41	-3.42	-0.23	-9.98	6.71
	5	alto	médio	-11.41	39.09	-18.96	-10.26	38.37	-29.36
C	1	muito baixo	médio	-1.03	1.28	-0.69	-5.80	1.74	-0.80
	2	muito baixo	muito alto	32.33	-71.32	-55.76	-10.01	25.36	73.13
	3	baixo	alto	1.01	-2.79	-4.65	5.08	1.91	2.97
	4	alto	alto	0.98	-11.77	2.88	14.85	-2.71	-1.65
	5	muito alto	baixo	1.07	1.66	-1.80	-1.29	-8.34	3.04
	6	muito alto	alto	-2.97	33.58	-40.14	-16.85	32.26	-2.43
	7	muito alto	muito alto	-0.48	19.16	-15.75	-4.85	14.33	-11.78

Conforme mostrado na tabela 5.11, todas as bases de regras resultantes são consistentes e, para os níveis A e B, estão próximas da base de regras do sistema nebuloso original (SNO).

A figura 5.18 ilustra os termos lingüísticos A_k^j , $k = 1, 2$ e $j = 1, \dots, 5$, definidos nos universos associados. Todas as partições que emergem da co-evolução atendem os critérios de visibilidade e, para os níveis A e B, estão bem próximas das partições do sistema original.

O mapeamento produzido pelo sistema nebuloso original (SNO) é ilustrado na figura 5.19(a). Após 1000 gerações o sistema nebuloso resultante de cada processo co-evolutivo produziu um mapeamento conforme mostrado nas figuras 5.19(b-d).

Os resultados mostrados nesta seção comprovam a eficiência do modelo proposto para encontrar elementos mais simples em espaços de busca complexos. Nota-se que, para os níveis A e B, a abordagem co-evolutiva foi capaz de produzir uma boa aproximação da curva original. Para

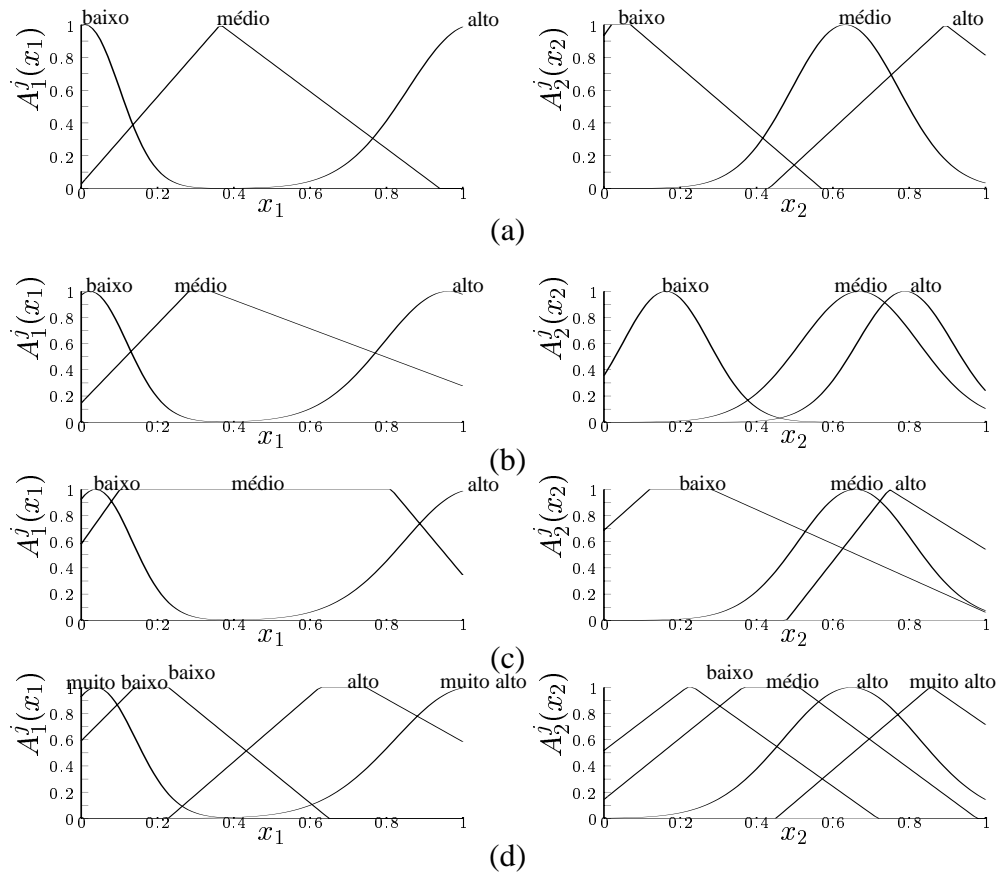


Figura 5.18: Partições nebulosas: (a) do sistema nebuloso original; (b) do sistema co-evolutivo para o nível A; (c) do sistema co-evolutivo para o nível B; (d) do sistema co-evolutivo para o nível C.

estes dois níveis, as pequenas discrepâncias entre as funções de pertinência e base de regras dos sistemas nebulosos resultantes, em relação à partição e base de regras do SNO, são compensadas pela flexibilidade dos consequentes não-lineares.

Para o nível C, o resultado, embora mais distante do original, mostra a grande capacidade de compressão do modelo, uma vez que com um máximo de 25 regras, o processo co-evolutivo convergiu para um sistema com apenas 7 regras. Os resultados deste nível mostram que a compressão das regras não foi acompanhada pela eliminação de termos desnecessários nas partições nebulosas, evidenciando a dependência do modelo em relação à granularidade máxima.

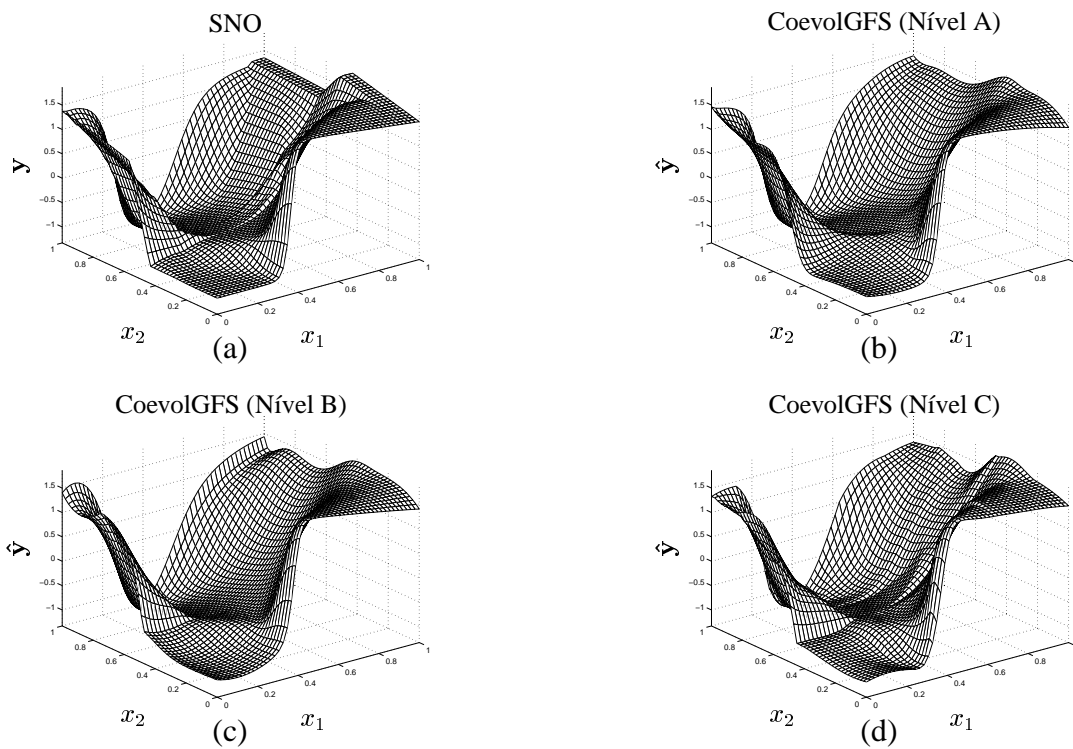


Figura 5.19: Mapeamento de Entrada-Saída produzido por: (a) sistema nebuloso original; (b) sistema co-evolutivo sujeito às restrições impostas no nível A; (c) sistema co-evolutivo sujeito às restrições impostas no nível B; (d) sistema co-evolutivo sujeito às restrições impostas no nível C.

Capítulo 6

Conclusões

A teoria da computação flexível está em busca contínua por dispositivos poderosos que representem e processem a informação. O objetivo maior é a obtenção de soluções finais com menor demanda por recursos computacionais. A principal consequência desta iniciativa é a substituição dos dispositivos de propósito geral por dispositivos mais dedicados, que utilizem heurísticas específicas para a definição dos atributos do sistema a ser modelado. Esta é uma condição necessária para que soluções parcimoniosas sejam obtidas na resolução de problemas complexos em inteligência computacional (Kearns & Vazirani, 1994). Quando a estrutura da solução final não é definida *a priori* e depende da natureza e complexidade do problema, os modelos resultantes se tornam dispositivos não-paramétricos (Härdle, 1990).

Algoritmos de aprendizado, para os quais tanto a estrutura dos sistemas nebulosos quanto os parâmetros do modelo são obtidos por treinamento baseados em dados de entrada-saída, constituem uma classe poderosa de dispositivos não-paramétricos. Toda vez que a complexidade do problema não pode ser determinada *a priori*, a possibilidade de ajuste da estrutura do sistema nebuloso deve ser explorada para se estimar a real flexibilidade do modelo de aproximação. A questão é como encontrar a configuração do sistema nebuloso que mapeie corretamente cada padrão de entrada no correspondente padrão de saída desejado.

Neste trabalho, a avaliação de desempenho das abordagens para o projeto automático de sistemas nebulosos foi medida com base em quatro aspectos distintos:

- acuidade da solução obtida;
- interpretabilidade da solução obtida;
- autonomia no processo de obtenção da solução;

- custo computacional para se implementar a solução obtida.

A abordagem proposta neste trabalho apresentou um modelo não-paramétrico baseado em co-evolução, com componentes de hierarquia e cooperação, para o projeto automático de sistemas nebulosos. As principais características da abordagem, são detalhadas a seguir:

- **co-evolução** - O processo co-evolutivo se baseou em populações de diferentes níveis hierárquicos representando espécies distintas, que codificaram soluções parciais do problema de projeto automático de sistemas nebulosos. A estrutura modular e hierárquica pôde ser implementada de modo que indivíduos em quatro níveis distintos - partições nebulosas (nível I), regras individuais (nível II), base de regras (nível III) e sistemas nebulosos (nível IV) - mantivessem uma relação cooperativa, e o *fitness* de um indivíduo dependesse do *fitness* de indivíduos pertencentes aos outros níveis.
- **hierarquia** - Indivíduos dos níveis superiores foram construídos a partir de indivíduos de níveis inferiores, ou seja, indivíduos de um nível mantêm relações de interdependências na instanciação de indivíduos de outros níveis. Relações hierárquicas também foram estabelecidas no cálculo do *fitness* dos indivíduos, nos diferentes níveis que evoluem.
- **cooperação** - O processo co-evolutivo induziu uma cooperação entre os indivíduos das diferentes espécies. A cooperação foi alcançada, tanto nas relações de interdependência dos indivíduos, quanto na etapa de avaliação do *fitness*.
- **acuidade** - A abordagem proposta produziu sistemas nebulosos acurados, capazes de aproximar funções com diferentes níveis de complexidade, com erro de treinamento e teste dentro de limites aceitáveis. A possibilidade da ocorrência de sobre-ajuste, resultante do uso de modelos extremamente flexíveis, como é o caso de modelos TS com conseqüentes não-lineares, foi atenuada através do algoritmo de poda proposto.
- **interpretabilidade** - Os sistemas nebulosos resultantes apresentaram bases de regras compreensíveis para um operador humano, formadas por um número reduzido de regras, todas consistentes, simplificadas pela presença de condições irrelevantes, e cujas partições nebulosas atenderam os critérios de visibilidade estipulados.
- **autonomia de projeto** - Parâmetros críticos dos sistemas nebulosos, como operadores de agregação dos antecedentes, número de regras nebulosas, tipo, formato e localização das funções de pertinência, emergiram do processo co-evolutivo, ao invés de serem definidos a

priori, tornando a autonomia de projeto mais próxima de ser alcançada. No entanto, alguns parâmetros continuam sendo arbitrados, como a granularidade máxima nos universos das variáveis associadas.

- **flexibilidade** - O uso de diferentes classes de funções no conseqüente e diferentes métodos de otimização dos parâmetros associados permitiu uma maior liberdade para se escolherem modelos com diferentes compromissos entre acuidade e interpretabilidade.
- **seleção de variáveis** - O uso de base de regras genéricas permitiu a seleção de quais variáveis são relevantes para cada regra individualmente. Além disso, é possível a realização de seleção de variáveis no âmbito geral da aplicação, embora as simulações realizadas não tenham evidenciado esta característica.
- **tratabilidade computacional** - As simulações mostraram que a abordagem co-evolutiva, embora demandando em média mais recursos computacionais do que as abordagens alternativas consideradas, foi capaz de manter a tratabilidade computacional no processo de obtenção da solução.
- **parcimônia** - A abordagem baseada em co-evolução produziu soluções finais bastante parcimoniosas. Com o paradigma co-evolutivo, houve uma tendência de se aumentar o custo de projeto com o objetivo de se reduzir o custo de implementação da solução final. Em muitas circunstâncias, o alto custo computacional de projeto é perfeitamente justificável, especialmente quando a solução resultante (modelo nebuloso para solucionar o problema) é parcimoniosa e acurada (Sober, 1975). Por exemplo, se o sistema nebuloso representa uma solução para um problema de engenharia, o dispositivo computacional correspondente será, provavelmente, implementado em larga escala na indústria. Portanto, no cálculo do custo final do produto manufaturado, o custo requerido para se implementar a solução (custo de produção do produto, que é **menor** para modelos não-paramétricos parcimoniosos) deverá prevalecer sobre o custo para se obter a solução (custo computacional no processo de modelagem), que é **maior** para modelos não-paramétricos parcimoniosos.

Todas as simulações realizadas consideraram uma única variável de saída, mas a extensão da abordagem para múltiplas saídas pode ser obtida de forma direta, tanto para o modelo Mamdani, quanto para o modelo Takagi-Sugeno. As simulações envolvendo problemas de aproximação de dados com alta confiabilidade, aproximação de dados sujeitos a ruídos, classificação de padrões e experimentos de caminho fechado mostraram a autonomia e flexibilidade do sistema para

encontrar modelos acurados, com grande capacidade de generalização, e que atenderam os critérios de interpretabilidade estabelecidos. Os melhores desempenhos obtidos, frente a um modelo neuro-nebuloso bastante poderoso, como é o caso do ANFIS, e a modelos obtidos por um método evolutivo tradicional, confirmaram a expectativa de que a co-evolução, associada à hierarquia e co-operação, representa uma alternativa promissora para o projeto automático de sistemas nebulosos.

6.1 Perspectivas Futuras

Embora a eficiência da abordagem proposta tenha sido comprovada através de inúmeros testes e simulações, algumas etapas ainda requerem experimentos e estudos mais avançados e deverão ser tratadas em trabalhos posteriores. Por exemplo, a questão relacionada à autonomia do projeto como um todo, incluindo os parâmetros dos SNs e do AG. A abordagem certamente diminui os parâmetros de projeto dos sistemas nebulosos, mas insere parâmetros importantes associados ao algoritmo genético proposto. Alguns testes foram realizados para se avaliarem diferentes taxas dos operadores genéticos, tipos de seleção, etc. Verificou-se que elevando-se o número máximo T de gerações para se atingir o critério de parada, a sensibilidade do processo evolutivo a esses parâmetros é reduzida. Entretanto, nenhuma análise quantitativa foi feita neste sentido. Existe portanto a necessidade de testes mais abrangentes, considerando diferentes condições iniciais para que resultados mais conclusivos possam ser derivados.

Outros pontos importantes a serem tratados em trabalhos futuros são:

- A implementação de um processo de agrupamento (*clustering*) para encontrar uma granularidade máxima e indicar a localização inicial das funções de pertinência mais apropriada para cada aplicação;
- Maior ênfase na interpretabilidade do modelo. Para isso, é necessária:
 - a implementação de um mecanismo de transformação de conseqüentes constantes (números reais) em conjuntos nebulosos, para aumentar a representação lingüística dos modelos TS com conseqüentes constantes;
 - a difusão do uso de modelos Mamdani, já que estes modelos são considerados mais interpretáveis do que os modelos Takagi-Sugeno;
- A realização de novas simulações envolvendo:

- casos para os quais não é possível se disponibilizar a saída desejada (aqui, os conseqüentes TS precisam ser evoluídos, visto que não há como se aplicar o método dos quadrados mínimos). Nestes casos, a escolha entre modelos Mamdani e TS poderia ser feita de forma automática, já que os modelos TS não teriam desempenho muito superior devido à ausência do processo de otimização dos conseqüentes;
- dados com distribuição não-uniforme (especialmente distribuições esparsas);
- problemas de controle e identificação de sistemas dinâmicos;
- Modificações no algoritmo genético proposto com o objetivo de:
 - reduzir o custo computacional para produzir a solução em universos envolvendo um número elevado de variáveis de entrada. Esta possibilidade depende da implementação de métodos de seleção de variáveis mais elaborados e eficientes, já que métodos de seleção envolvendo buscas exaustivas podem ser impraticáveis para um grande número de variáveis (Hall, 1998);
 - testar outras técnicas para implementar o elitismo e a reparação;
 - testar outros operadores genéticos (por exemplo: crossover aritmético);
 - incorporar outros métodos de busca local, além do que já é realizado pela otimização dos conseqüentes TS;
 - implementar novas configurações hierárquicas que permitam, inclusive, incluir as classes de funções e os métodos de otimização dos conseqüentes como parâmetros a serem evoluídos;
 - adaptar a técnica proposta a outras estruturas de processamento de informação;
- Modificações na etapa de otimização dos conseqüentes para avaliar outros algoritmos de eliminação de redundância em comparação com o algoritmo de poda proposto;
- Inclusão dos operadores do mecanismo de inferência e restrições na partição para a abordagem EGFS (não emprega co-evolução), para que resultados mais conclusivos possam ser extraídos a respeito das vantagens do uso da co-evolução.
- Modelagem e solução do projeto automático de sistemas nebulosos por programação multi-objetivo. Este tópico já tem sido trabalhado e os resultados preliminares foram apresentados em Delgado *et al.* (2002a,b);

- A implementação de um pacote de software (*toolbox*) para Matlab ou outras linguagens de programação (Java .

Glossário

Acrônimos

- AE** Algoritmo Evolutivo, pág. 13.
- AG** Algoritmo Genético, pág. 6.
- ANFIS** *Adaptive Neural Fuzzy Inference Systems*, pág. 5.
- BR** Base de Regras, pág. 89.
- BRE** Base de Regras Específicas, pág. 106.
- BRG** Base de Regras Genéricas, pág. 106.
- CE** Computação Evolutiva, pág. 1.
- CF** Computação Flexível, pág. 1.
- CoA** Centro de Área, pág. 71.
- CoevolGFS** *Coevolutionary Genetic Fuzzy Systems*, pág. 116.
- CoG** Centro de Massa, pág. 71.
- EE** Estratégia Evolutiva, pág. 11.
- EGFS** *Evolutionary Genetic Fuzzy Systems*, pág. 116.
- EQM** Erro Quadrático Médio, pág. 83.
- EQM_r** Erro Quadrático Médio calculado para dados com ruído, pág. 132.
- FP** Função de Pertinência, pág. 89.

- GFS** *Genetic Fuzzy Systems*, pág. 6.
- GranulMax** Granularidade Máxima do universo, pág. 119.
- HGFS** *Hierarchical Genetic Fuzzy Systems*, pág. 87.
- LD** Linearmente Dependente, pág. 110.
- LI** Linearmente Independente, pág. 111.
- MaxReg** Máximo de Regras em uma base de regras, pág. 120.
- MLP-BckPr** *Multi Layer Perceptron* treinada por *Brackpropagation*, pág. 117.
- MoM** Média dos Máximos, pág. 70.
- MP** *Modus Ponens*, pág. 63.
- OWA** Operador de Média Ponderada Ordenada, pág. 54.
- PE** Programação Evolutiva, pág. 11.
- PG** Programação Genética, pág. 22.
- PN** Partição Nebulosa, pág. 104.
- RCOND** Estimador de condição recíproca de uma matriz, pág. 111.
- RN** Redes Neurais, pág. 2.
- SN** Sistema Nebuloso, pág. 2.
- SNO** Sistema Nebuloso Original, pág. 146.
- TS** Takagi-Sugeno, pág. 7.

Símbolos

- \wedge Operador de conjunção, pág. 50.
- \vee Operador de disjunção, pág. 50.

- # condição irrelevante (*don't care*) nas proposições do antecedente ou conseqüente da regra nebulosa, pág. 63.
- Operador da composição sup-t entre duas relações nebulosas quaisquer, pág. 56.
- Operador da composição inf-s entre duas relações nebulosas quaisquer, pág. 57.
- \mathfrak{R}^j Espaço j -dimensional dos dados entrada-saída, pág. 107.
- β_j Resultado normalizado da agregação do antecedente na regra j , pág. 108.
- γ Grau mínimo de sobreposição entre as funções de pertinência na partição do universo, pág. 83.
- κ Grau mínimo de sobreposição entre as funções de pertinência na partição do universo, pág. 83.
- λ_j vetor que armazena as agregações normalizadas e entradas, pág. 108.
- μ Média na distribuição normal (dados com ruído), pág. 131.
- $\mu_A(x)$ Função de pertinência de um conjunto nebuloso A definido no universo \mathbf{X} , pág. 44.
- $\mu_{(\overline{A})}(x)$ Função de pertinência do complemento \overline{A} de um conjunto nebuloso A definido em \mathbf{X} , pág. 49.
- $\mu_{(A \cap B)}(x)$ Função de pertinência da interseção $A \cap B$ de dois conjuntos nebulosos A e B definidos em \mathbf{X} , pág. 49.
- $\mu_{(A \cup B)}(x)$ Função de pertinência da união $A \cup B$ de dois conjuntos nebulosos A e B definidos em \mathbf{X} , pág. 49.
- μ_j Resultado da agregação do antecedente na regra j , pág. 73.
- μ_{m_k} *Matching* entre cada proposição atômica (associada à variável k) do antecedente e a proposição atômica na proposição de entrada para uma regra qualquer, pág. 73.
- ξ Realização independente dos dados com ruído, pág. 130.
- σ Desvio padrão na distribuição normal (dados com ruído), pág. 131.

- σ_k Dispersão na função de pertinência Gaussiana, pág. 97.
- χ Variável aleatória com distribuição uniforme no intervalo $[0, 1]$, pág. 25.
- $\Delta(\cdot)$ Função que depende da geração t e dos limites do domínio do alelo e define o incremento (ou decremento) da mutação real não-uniforme., pág. 30.
- Λ Matriz dos coeficientes do sistema de equação linear $\hat{y} = \Lambda w$, pág. 109.
- Ξ Conjunto formado por sub-conjuntos de padrões de treinamento com ruído, pág. 130.
- Φ índice de todas as colunas da matriz Λ , pág. 112.
- Ψ_j matriz de ponderação para o problema dos quadrados mínimos ponderados, pág. 110.
- Ω Região compacta na qual estão definidos os mapeamentos a serem aproximados, pág. 106.
- A' Proposição de entrada (fato) utilizada no *modus ponens* generalizado ou no raciocínio nebuloso, pág. 63.
- A_k^j termo lingüístico associado à k -ésima variável de entrada na j -ésima regra da base de regras, pág. 71.
- $A \rightarrow B$ Conjunção nebulosa (grânulo) envolvendo os conjuntos A e B , pág. 67.
- $A \Rightarrow B$ Implicação nebulosa envolvendo os conjuntos A e B , pág. 67.
- A_a Operador de agregação dos antecedentes da regra nebulosa, pág. 62.
- A_c Operador de agregação dos consequentes da regra nebulosa, pág. 62.
- A_R Operador de agregação das regras que compõem uma base de regras, pág. 68.
- Alf_k** Conjunto de valores possíveis que podem ser atribuídos ao k -ésimo alelo de um indivíduo qualquer, pág. 22.
- b_k Alelo ou bit no *locus* k de um cromossomo com codificação binária, pág. 33.
- b_{1k} Extremo esquerdo (parâmetro a) da função trapezoidal, pág. 97.
- b_{2k} Extremo direito (parâmetro b) da função trapezoidal, pág. 97.

- B' Proposição de saída (conclusão) extraída no *modus ponens* generalizado ou no raciocínio nebuloso, pág. 63.
- B_k^j termo lingüístico associado à k -ésima variável de saída na j -ésima regra da base de regras, pág. 71.
- C_{α_A} Elementos do universo para os quais os graus de pertinência são superiores ou iguais a α , pág. 47.
- C_{1k} Alelo que define a distância da função de pertinência para a anterior, pág. 97.
- C_{2k} Alelo que define a abertura central ($n - m$) da função trapezoidal, pág. 97.
- Cl Rótulo que define a classe nos problemas de classificação de padrões, pág. 140.
- cilA** Extensão cilíndrica, pág. 65.
- D_k Domínio $D_k = [l_k, u_k]$ que define os valores mínimo (l_k) e máximo (u_k) que um alelo no *locus* k pode assumir, pág. 22.
- Disp_k^{max}** Dispersão máxima definida para a função de pertinência Gaussiana, pág. 97.
- EAI_{pmut}** Escolha aleatória do ponto de mutação, pág. 101.
- EFi_{pmut}** Escolha do ponto de mutação baseada no *fitness*, pág. 101.
- f Semântica da regra nebulosa, pág. 60.
- f_i Semântica da regra nebulosa dada por uma implicação (norma-i), pág. 60.
- f_l Semântica da regra (conjunção) definida por Larsen (norma-t=produto algébrico), pág. 61.
- f_m Semântica da regra (conjunção) definida por Mamdani (*norma - t=min*), pág. 61.
- f_t Semântica da regra nebulosa dada por uma conjunção (norma-t), pág. 60.
- f_s Semântica da regra nebulosa dada por uma disjunção (norma-s), pág. 60.
- F Função de avaliação desempenho (ou *fitness*), pág. 14.
- $F(h_i^t)$ Medida de desempenho (ou *fitness*) do indivíduo i na geração t , pág. 14.

- $\mathcal{F}(X)$ Família de conjuntos nebulosos definidos no universo X , pág. 58.
- g Função no conseqüente da regra j para os modelos TS, pág. 77.
- g_j^t Coleção j de genótipos na geração t , definida no espaço genotípico G , pág. 16.
- G Espaço genotípico, pág. 15.
- G_k Número máximo de termos lingüísticos para a k -ésima variável de entrada, pág. 102.
- \mathcal{G} Gramática para a geração dos termos ou rótulos, pág. 58.
- h_i^t Indivíduo i na geração t , pág. 13.
- H_A Altura (valor máximo da função de pertinência) do conjunto nebuloso A , pág. 47.
- I Interseção na regra composicional de inferência, pág. 65.
- \mathbf{I}_{alf} Alfabeto de entrada no espaço genótipo \times fenótipo, pág. 15.
- L_k Alelo que define abertura do lado esquerdo ($m - a$) da função trapezoidal, pág. 97.
- L_{prun} Limiar do algoritmo de poda, pág. 111.
- m Total de regra numa base de regras, pág. 68.
- m_1 Mapeamento que define a epigênese, pág. 16.
- m_2 Mapeamento que define a seleção e migração de indivíduos dentro da população local, pág. 16.
- m_3 Mapeamento que define a representação ou sobrevivência genotípica, pág. 17.
- m_4 Mapeamento que define alterações genéticas (mutação e crossover), pág. 17.
- $m1k$ Média esquerda (parâmetro m) da função trapezoidal, pág. 97.
- $m2k$ Média direita (parâmetro n) da função trapezoidal, pág. 97.
- \mathcal{M} Regra que associa um rótulo a um conjunto nebuloso, pág. 58.
- n Total de variáveis de entrada, pág. 108.

- N Total de padrões de entrada, pág. 83.
- N_A Núcleo (elementos com grau de pertinência igual a 1) do conjunto nebuloso A , pág. 47.
- \mathcal{N} Distribuição normal dos dados com ruído, pág. 131.
- p Índice do padrão de entrada, pág. 83.
- p_t Parâmetro das normas triangulares: norma-t e norma-s, pág. 52.
- p_j^t Coleção j de fenótipos na geração t , definida no espaço fenotípico \mathbf{P} , pág. 16.
- \mathbf{P} Espaço fenotípico, pág. 15.
- P_a Proposição (conjunção) no antecedente da regra nebulosa, pág. 62.
- P_c Proposição (conjunção) no conseqüente da regra nebulosa, pág. 62.
- P_C Probabilidade de crossover, pág. 27.
- P_i Proposição de entrada utilizada pelo mecanismo de inferência, pág. 71.
- P_M Probabilidade de mutação, pág. 29.
- P_o Proposição de saída inferida pelo mecanismo de inferência, pág. 71.
- P_{oj} Proposição de saída inferida na regra j , pág. 73.
- P^t População na geração t , pág. 13.
- P_{sel}^i Probabilidade do indivíduo i ser selecionado para compor a população na próxima geração, pág. 24.
- $P_A(x)$ Função de pertinência de um conjunto A qualquer, pág. 44.
- proj** $_X$ Projeção no universo X , pág. 65.
- Q Dimensão do vetor de parâmetros w_j , pág. 77.
- Q_a Proposição (disjunção) no antecedente da regra nebulosa, pág. 62.
- Q_c Proposição (disjunção) no conseqüente da regra nebulosa, pág. 62.

- r** Índice das colunas LI de Λ , pág. 111.
- r_j realização da variável aleatória χ_j com distribuição uniforme no intervalo $[0, 1]$, pág. 30.
- R Relação ou regra nebulosa, pág. 54.
- R_j j -ésima regra nebulosa em uma base de regras, pág. 68.
- R_k Alelo que define abertura do lado direito ($b - n$) da função trapezoidal (distância para o valor de referência), pág. 97.
- s Tamanho do cromossomo que representa um indivíduo qualquer na população, pág. 22.
- s_j j -ésima norma-s, pág. 52.
- S_A Suporte (elementos com grau de pertinência maior do que zero) do conjunto nebuloso A , pág. 47.
- S_k Alelo que define o tipo da função de pertinência, pág. 97.
- S_p Total de indivíduos na população P^t , pág. 13.
- t_j j -ésima norma-t, pág. 50.
- T Limite máximo de gerações, pág. 30.
- $\mathcal{T}(X)$ Conjunto dos termos lingüísticos da variável lingüística X , pág. 58.
- u_k Posições arbitrárias para as entradas não-nebulosas associadas à variável k , pág. 76.
- v_k^t Alelo no *locus* k de um indivíduo qualquer na geração t , pág. 29.
- v_{ik}^t Alelo que representa o gene no *locus* k do indivíduo i na geração t , pág. 22.
- w_j Vetor dos parâmetros da função g no conseqüente da regra j para os modelos TS, pág. 77.
- w^* Conjunto ótimo de todos os parâmetros no conseqüente, pág. 108.
- x Vetor das variáveis base representando as variáveis de entrada, pág. 78.
- X Variável lingüística definida no universo \mathbf{X} , cuja variável base é x , pág. 58.

- X** Universo de um conjunto qualquer, pág. 43.
- X_k k -ésima variável de entrada (variável lingüística), pág. 71.
- y_p Saída desejada para o p -ésimo padrão de entrada, pág. 83.
- \hat{y}_p Saída estimada para o p -ésimo padrão de entrada, pág. 83.
- Y_k k -ésima variável de saída, pág. 71.
- Z** Matriz de entrada do algoritmo de poda, cujas colunas LD serão eliminadas, pág. 111.
- Z** Conjunto dos números inteiros, pág. 29.

Bibliografia

- Aarts, E. & Korst, J. (1989). *Simulated Annealing and Boltzman Machines - A stochastic Approach to Combinatorial Optimization and Neural Computing*. John Wiley & Sons.
- Atmar, W. (1994). 'Notes on the Simulation of Evolution'. *IEEE Transactions on Neural Networks*, Vol. 5, N. 1, pp 130–148.
- Bäck, T., Fogel, D. B., & Michalewicz, Z., editors (2000a). *Evolutionary Computation 1: Basic Algorithms and Operators*. Institute of Physics Publishing.
- Bäck, T., Fogel, D. B., & Michalewicz, Z., editors (2000b). *Evolutionary Computation 2: Advanced Algorithms and Operators*. Institute of Physics Publishing.
- Berthold, M. & Hand, D. J., editors (1999). *Intelligent Data Analysis: An Introduction*. Springer-Verlag.
- Castellano, G. & Fanelli, A. M. (2000). 'A staged approach for generation and compression of fuzzy classification rules'. In *Proceedings of FUZZ-IEEE'2000*, San Antonio, USA. pp 42–47.
- Chen, M.-S. & Liou, R. (1999). 'An Efficient Learning Method of Fuzzy Inference System'. In *Proceedings of FUZZ-IEEE'99*, Seoul, Korea. pp 634–638.
- Cherkassky, V., Gehring, D., & Mulier, F. (1996). 'Comparison of Adaptive Methods for Function Estimation from Samples'. *IEEE Transactions on Neural Networks*, Vol. 7, N. 4, pp 969–984.
- Cordón, O., del Jesus, M. J., Herrera, F., & Villar, P. (2001a). 'A Multiobjective Genetic Algorithm for Feature Selection and Granularity Learning in Fuzzy-Rule-Based Classification Systems'. In *Proceedings of 9thIFSA World Congress and 20thNAFIPS International Conference*, Vancouver, CA. pp 1253–1258.

- Cordón, O. & Herrera, F. (1999). 'A two-stage evolutionary process for designing TSK fuzzy rule-based systems'. *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 29, N. 6, pp 703–714.
- Cordón, O., Herrera, F., Gomide, F., Hoffmann, F., & Magdalena, L. (2001b). 'Ten Years of Genetic Fuzzy Systems: Current Framework and New Trends'. In *Proceedings of 9thIFSA World Congress and 20thNAFIPS International Conference*, Vancouver, CA. pp 1241–1246.
- Cordón, O., Herrera, F., Hoffmann, F., & Magdalena, L. (2001c). *Genetic Fuzzy Systems. Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*. Advances in Fuzzy Systems - Applications and Theory. World Scientific.
- Davis, L., editor (1987). *Genetic Algorithms and Simulated Annealing*. Morgan Kauffmann Publishers.
- De Jong, K. A. (1975). *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, Department of Computer and Communications, University of Michigan.
- Delgado, M., Von Zuben, F., & Gomide, F. (1999). 'Modular and Hierarchical Evolutionary Design of Fuzzy Systems'. In *Proceedings of the Genetic and Evolutionary Computation Conference - GECCO'99*, Orlando, USA. pp 180–187.
- Delgado, M. R., Von Zuben, F., & Gomide, F. (2000a). 'Evolutionary design of Takagi-Sugeno fuzzy systems: a modular and hierarchical approach'. In *Proceedings of the Ninth IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, San Antonio, USA. pp 447–452.
- Delgado, M. R., Von Zuben, F., & Gomide, F. (2000b). 'Optimal Parameterization of Evolutionary Takagi-Sugeno Fuzzy Systems'. In *Proceedings of 8th Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference (IPMU)*, Madrid, Spain. pp 650–657.
- Delgado, M. R., Von Zuben, F., & Gomide, F. (2001a). 'Hierarchical Genetic Fuzzy Systems'. *Information Sciences - Special Issue on Recent Advances in Genetic Fuzzy Systems*, Vol. 136, N. 1–4, pp 29–52.
- Delgado, M. R., Von Zuben, F., & Gomide, F. (2001b). 'Local and Global Estimation of Takagi-Sugeno Consequent Parameters in Genetic Fuzzy Systems'. In *Proceedings of 9thIFSA World Congress and 20thNAFIPS International Conference*, Vancouver, CA. pp 1247–1252.

- Delgado, M. R., Von Zuben, F. J., & Gomide, F. (2002a). 'Hierarchical Genetic Fuzzy Systems: Accuracy, Interpretability and Design Autonomy'. In *Trade-off Between Accuracy and Interpretability in Fuzzy Rule-Based Modeling - Studies in Fuzziness and Soft Computing (In Press)*. Physica-Verlag.
- Delgado, M. R., Von Zuben, F. J., & Gomide, F. (2002b). 'Multi-Objective Decision Making: Towards Improvement of Accuracy, Interpretability and Design Autonomy in Hierarchical Genetic Fuzzy Systems'. In *Proceedings of 2002 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) - Aceito.*, Honolulu, HI.
- Dubois, D. & Prade, H. (1980). *Fuzzy Sets and Systems: Theory and Applications*. Academic Press.
- Dyckhoff, H. & Pedrycz, W. (1984). 'Generalized Mean as a Model of Compensative Connectives'. *Fuzzy Sets and Systems*, Vol. 14, pp 143–154.
- Ehrlich, P. R. & Raven, P. H. (1964). 'Butterflies and Plants: a Study in coevolution'. *Evolution*, Vol. 18, pp 586–608.
- Espinosa, J. & Vandewalle, J. (2000). 'Constructing Fuzzy Models with Linguistic Integrity from Numerical Data-AFRELI Algorithm'. *IEEE Transactions on Fuzzy Systems*, Vol. 8, N. 5, pp 591–600.
- Fisher, R. A. (1936). 'The Use of Multiple Measurements in Taxonomic Problems'. In *Annual Eugenics*, volume 7. pp 179–188.
- Fogel, D. B. (1995). *Evolutionary Computation - Toward a New Philosophy of Machine Intelligence*. IEEE Press.
- Fogel, D. B., Owens, A. J., & Walsh, M. J. (1966). *Artificial Intelligence through Simulated Evolution*. John Wiley.
- Glover, F. & Laguna, M. (1997). *Tabu Search*. Kluwer Academic Publishers.
- Golub, G. H. & Van Loan, C. F. (1996). *Matrix Computations*. Johns Hopkins Series in the Mathematical Sciences. Johns Hopkins University Press, 3rd edition.
- Gonzales, A. & Perez, R. (1998). 'Completeness and Consistency Conditions for Learning Fuzzy Rules'. *Fuzzy Sets and Systems*, Vol. 96, pp 37–51.

- Hall, M. (1998). *Correlation-based Feature Selection for Machine Learning*. PhD thesis, Hamilton, NZ: Waikato University, Department of Computer Science.
- Hang, X. Z. C.-C. (1997). 'The Min-Max Function Differentiation and Training of Fuzzy Neural Networks'. *IEEE Transactions on Neural Networks*, Vol. 7, N. 5, pp 1139 – 1150.
- Hartl, D. L. & Clark, A. G. (1989). *Principles of Population Genetics*. Sinauer.
- Haykin, S. (1999). *Neural Networks - A Comprehensive Foundation*. Macmillan, 2nd edition.
- Heider, H. & Drabe, T. (1997). 'A Cascaded Genetic Algorithm for Improving Fuzzy Systems Design'. *International Journal of Approximate Reasoning*, Vol. 17, N. 4, pp 351–368.
- Herrera, F. & Magdalena, L. (1998). 'Introduction: Genetic Fuzzy Systems'. *International Journal of Intelligent Systems*, Vol. 13, pp 887–890.
- Hillis, W. D. (1990). 'Co-Evolving Parasites Improve Simulated Evolution as an Optimization Procedure'. *Physica D*, Vol. 42, N. 1–3, pp 228–234.
- Hoffmann, F. & Nelles, O. (2000). 'Structure identification of TSK-fuzzy systems using genetic programming'. In *Proceedings of IPMU 2000*, Madrid, Spain. pp 438–445.
- Hoffmann, F. & Pfister, G. (1995). 'A New Learning Method for the Design of Hierarchical Fuzzy Controllers Using Messy Genetic Algorithms'. In *Proceedings of the 6th IFSA World Congress*, São Paulo, Brazil. pp 249–252.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.
- Homaifar, A. & McCormick, E. (1995). 'Simultaneous Design of Membership Functions and Rule Sets for Fuzzy Controller Using Genetic Algorithms'. *IEEE Transactions on Fuzzy Systems*, Vol. 3, N. 2, pp 129–139.
- Hornik, K., Stinchcombe, M., & White, H. (1989). 'Multilayer Feedforward Networks are Universal Approximators'. *Neural Networks*, Vol. 2, N. 5, pp 359–366.
- Härdle, W. (1990). *Applied Nonparametric Regression*. Cambridge University Press.
- Ishibuchi, H. & Nakashima, T. (1999). 'Genetic-Algorithm-Based Approach to Linguistic Approximation of Nonlinear Functions with Many Input Variables'. In *Proceedings of FUZZ-IEEE'99*, Seoul, Korea. pp 779–784.

- Ishibuchi, H., Takeuchi, D., & Nakashima, T. (2001). 'GA-based Approaches to Linguistic Modeling of Nonlinear Functions'. In *Proceedings of 9thIFSA World Congress and 20thNAFIPS International Conference*, Vancouver, CA.
- Iyoda, E. M. (2000). Inteligência computacional no projeto automático de redes neurais híbridas e redes neurofuzzy heterogêneas. Dissertação de Mestrado, Faculdade de Engenharia Elétrica e de Computação, Unicamp, Campinas, SP, Brasil.
- Iyoda, E. M. & Von Zuben, F. J. (2002). 'Hybrid Neural Networks: An Evolutionary Approach with Local Search'. *Integrated Computer-Aided Engineering*, Vol. 9, N. 1, pp 57 – 72.
- Jang, J.-S. (1993). 'ANFIS: Adaptive-Network-based Fuzzy Inference Systems'. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, N. 3, pp 665–685.
- Jang, J.-S., Sun, C.-T., & Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing*. Prentice Hall.
- Jin, Y. (2000). 'Fuzzy Modeling of High-Dimensional Systems: Complexity Reduction and Interpretability Improvement'. *IEEE Transactions on Fuzzy Systems*, Vol. 8, N. 2, pp 212–221.
- Juillé, H. & Pollack, J. B. (1996). 'Co-evolving Intertwined Spirals'. In *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, San Diego, USA. pp 461–468.
- Karr, C. L. (1991a). 'Applying Genetics to Fuzzy Logic'. *AI Expert*, Vol. 6, N. 3, pp 38–43.
- Karr, C. L. (1991b). 'Genetic Algorithms for Fuzzy Controllers'. *AI Expert*, Vol. 6, N. 2, pp 26–33.
- Kearns, M. J. & Vazirani, U. V. (1994). *An Introduction to Computational Learning Theory*. MIT Press.
- Klement, E. P., Mesiar, R., & Pap, E. (2000). *Triangular Norms*. Kluwer Academic Publishers.
- Kosko, B. (1997). *Fuzzy Engineering*. Prentice Hall.
- Koza, J. R. (1992a). *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press.
- Koza, J. R. (1992b). *Genetic Programming: On the Programming of Computers by means of Natural Selection*. MIT Press.

- Koza, J. R., Andre, D., Bennett, F. H., & Keane, M. A. (1998). *Genetic Programming III: Automatic Programming and Automatic Circuit Synthesis*. Morgan Kaufmann.
- Lang, K. J. & Witbrock, M. J. (1988). 'Learning to tell two spirals apart'. In Touretzky, D., Hinton, G., & Sejnowski, T., editors, *Proc. 1988 Connectionist Models Summer School*, San Mateo CA. Morgan Kaufmann, pp 52–59.
- Larsen, P. M. (1980). 'Industrial Applications of Fuzzy Logic Control'. *International Journal of Man-Machine Studies*, Vol. 12, N. 1, pp 3–10.
- Lee, M. A. & Takagi, H. (1993). 'Integrating Design Stages of Fuzzy Systems using Genetic Algorithms'. In *Proceedings of FUZZ-IEEE'93*, San Francisco, USA. pp 612–617.
- Lin, C.-T. & Lee, C. S. G. (1996). *Neural Fuzzy Systems*. Prentice Hall.
- Lukasiewicz, J. (1970). 'Philosophical Remarks on Many-Valued Systems of Propositional Logic'. In *Studies in Logic and the Foundations of Mathematics*. North Holland.
- Mamdani, E. H. & Assilian, S. (1975). 'An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller'. *International Journal of Man-Machine Studies*, Vol. 7, N. 1, pp 1–13.
- Manual Matlab (1999). *MATLAB The Language of Technical Computing*. The MathWorks, Inc., version 5.3 edition.
- Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag.
- Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. MIT Press.
- Moriarty, D. E. & Miikkulainen, R. (1997). 'Forming Neural Networks Through Efficient and Adaptive Coevolution'. *Evolutionary Computation*, Vol. 5, N. 4, pp 373–399.
- Moriarty, D. E. & Miikkulainen, R. (1998). 'Hierarchical Evolution of Neural Networks'. In *Proceedings of the IEEE International Conference on Evolutionary Computation*, Alaska, USA. pp 428–433.
- Moscato, P. (1999). 'Memetic Algorithms: A Short Introduction'. In Corne, D., M. D., & Glover, F., editors, *New Ideas in Optimization*, pages 219 – 234. McGraw Hill.

- Nauck, D. & Kruse, R. (1999). 'Neuro-Fuzzy Systems for Function Approximation'. *Fuzzy Sets and Systems*, Vol. 101, pp 261–271.
- Nomura, H., Hayashi, I., & Wakami, N. (1991). 'A self-tuning Method of Fuzzy Control by Descent Method'. In *Proceedings of the International Fuzzy Systems Association*, Bruxelles, Belgium. pp 155–158.
- Nozaki, K., Ishibuchi, H., & Tanaka, H. (1997). 'A simple but powerful heuristic method for generating fuzzy rules from numerical data'. *Fuzzy Sets and Systems*, Vol. 86, pp 251–270.
- Olsson, B. (2001). 'Co-evolutionary Search in Asymmetric Spaces'. *Information Sciences*, Vol. 133, pp 103–125.
- Pedrycz, W. & Gomide, F. (1998). *An Introduction to Fuzzy Sets: Analysis and Design*. MIT Press, Cambridge.
- Potter, M. A. & De Jong, K. A. (2000). 'Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents'. *Evolutionary Computation*, Vol. 8, N. 1, pp 1–29.
- Rechenberg, I. (1965). Cybernetic solution path of an experimental problem. Technical report, Royal Aircraft Establ., Farnborough, UK.
- Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien Biologischer Evolution*. Frommann-Holzboog.
- Resher, N. (1969). *Many-Valued Logic*. Mc Graw Hill.
- Rizzi, A., Frattale, F. M., & Martinelli, G. (1999). 'Automatic training of ANFIS networks'. In *Proceedings of FUZZ-IEEE'99*, Seoul, Korea. pp 1655–1660.
- Rocha, A. F. (1992). *Neural Nets: A Theory of Brains and Machines*. Lecture Notes in Artificial Intelligence. Springer-Verlag.
- Rojas, I., Pomares, H., Fernandez, F. J., Bernier, J. L., Pelayo, F. J., & Pietro, A. (1999). 'A New Methodology to Obtain Fuzzy Systems Autonomously from Training Data'. In *Proceedings of IEEE Intern Fuzzy System Conference*, Seoul, Korea. pp 527–532.
- Rosin, C. & Belew, R. K. (1997). 'New Methods for Competitive Coevolution'. *Evolutionary Computation*, Vol. 5, N. 1, pp 1–29.

- Roughgarden, J. (1976). 'Resource Partitioning Among Competing Species: a Coevolutionary Approach'. *Theoretical Population Biology*, Vol. 9, N. 3, pp 388–424.
- Sareni, B. & Krähenbühl, L. (1998). 'Fitness Sharing and Niching Methods Revisited'. *IEEE Transactions on Evolutionary Computation*, Vol. 2, N. 3, pp 97–106.
- Schwefel, H.-P. (1981). *Numerical Optimization of Computer Models*. John Wiley.
- Setnes, M., Babuška, R., & Verbruggen, H. B. (1998). 'Rule-Based Modeling: Precision and Transparency'. *IEEE Transactions on Systems, Man, and Cybernetics–Part C*, Vol. 28, N. 1, pp 165–169.
- Shi, Y., Eberhart, R., & Chen, Y. (1998). 'Evolutionary Modular Fuzzy System'. In *Proceedings of International Conference on Evolutionary Computation*, Alaska, USA. pp 428–433.
- Shi, Y., Eberhart, R., & Chen, Y. (1999). 'Implementation of Evolutionary Fuzzy Systems'. *IEEE Transactions on Fuzzy Systems*, Vol. 7, N. 2, pp 109–119.
- Siarry, P. & Guely, F. (1998). 'A Genetic Algorithm for Optimizing Takagi-Sugeno Fuzzy Rule Bases'. *Fuzzy Sets and Systems*, Vol. 99, N. 1, pp 37–47.
- Simpson, P. K. (1992). 'Fuzzy Min-Max Neural Networks- Part I: Classification'. *IEEE Transactions on Neural Networks*, Vol. 3, N. 5, pp 776–786.
- Sober, E. (1975). *Simplicity*. Oxford University Press.
- Steels, L. (1995). 'When are Robots Intelligent Autonomous Agents?'. *Journal of Robotics and Autonomous Systems*, Vol. 15, pp 3–9.
- Takagi, H. & Lee, M. (1993). 'Neural Networks and Genetic Algorithms approaches to auto design of fuzzy systems'. In *Proceedings of Fuzzy Logic in Artificial Intelligence*, Linz, Austria. pp 68 – 79.
- Takagi, T. & Sugeno, M. (1983). 'Derivation of Fuzzy Control Rules from Human Operator's Control Actions'. In *Proceedings of the IFAC Symp. on Fuzzy Information, Knowledge Representation and Decision Analysis*, Marseilles, France. pp 55–60.
- Thrift, P. (1991). 'Fuzzy Logic Synthesis with Genetic Algorithms'. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, San Diego, USA. pp 509–513.

- Tsukamoto, Y. (1979). 'An Approach to Fuzzy reasoning Methods'. In Ragade, R. & Yager, R., editors, *Advances in Fuzzy Set Theory*, pages 137–149. Amsterdam: North-Holland.
- Von Zuben, F. J. (2000). *Computação evolutiva: Uma abordagem pragmática*. Tutorial: Notas de Aula da disciplina IA707, Faculdade de Engenharia Elétrica e de Computação - Universidade Estadual de Campinas.
- Vose, M. D. (1991). 'Generalizing the Notion of Schema in Genetic Algorithms'. *Artificial Intelligence*, Vol. 50, pp 385–396.
- Wineberg, M. (2000). *Improving the Behavior of the Genetic Algorithm in a Dynamic Environment*. PhD thesis, School of Computer Science - Carleton University, Ottawa, Canada.
- Wolpert, D. H. & Macready, W. G. (1997). 'No Free Lunch Theorems for Optimization'. *IEEE Transactions on Evolutionary Computation*, Vol. 1, pp 67–82.
- Yager, R. (1988). 'On Ordered Weighted Averaging Aggregation Operations in Multicriteria Decision Making'. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 18, pp 183–190.
- Yao, X. (1993). 'A Review of Evolutionary Artificial Neural Networks'. *International Journal of Intelligent Systems*, Vol. 4, pp 203 – 222.
- Yen, J., Wang, L., & Gillespie, C. W. (1998). 'Improving the Interpretability of TSK Fuzzy Models by Combining Global Learning and Local Learning'. *IEEE Transactions on Fuzzy Systems*, Vol. 6, N. 4, pp 530–537.
- Zadeh, L. A. (1965). 'Fuzzy Sets'. *Information and Control*, Vol. 8, pp 338–352.
- Zadeh, L. A. (1975). 'The Concept of Linguistic Variable and its Application to Approximate Reasoning'. *Information Sciences*, Vol. 8, pp 199–249 (part I), 301–357 (partII).
- Zimmermann, H. & Zysno, P. (1980). 'Latent Connectives in Human Decision Making'. *Fuzzy Sets and Systems*, Vol. 4, pp 37–51.

Índice de Autores

- Aarts & Korst (1989), 13
Atmar (1994), 15, 17
Bäck *et al.* (2000a), 1, 12, 27, 29, 102
Bäck *et al.* (2000b), 1, 12, 27, 102
Berthold & Hand (1999), 2, 12, 43, 58, 59,
129, 130
Castellano & Fanelli (2000), 59, 118, 139, 143,
144
Chen & Liou (1999), 6, 87
Cherkassky *et al.* (1996), 136
Cordón *et al.* (2001c), 6
Cordón *et al.* (2001b), 82
Cordón *et al.* (2001a), 59
Cordón & Herrera (1999), 3, 5
Davis (1987), 23
De Jong (1975), 26
Delgado *et al.* (2000a), 5, 78, 91
Delgado *et al.* (2000b), 7, 109, 111
Delgado *et al.* (2001b), 82
Delgado *et al.* (2001a), 3, 6, 8, 59, 84, 87, 90,
96
Delgado *et al.* (2002a), 155
Delgado *et al.* (2002b), 155
Delgado *et al.* (1999), 6, 105
Dubois & Prade (1980), 54
Dyckhoff & Pedrycz (1984), 54
Ehrlich & Raven (1964), 35
Espinosa & Vandewalle (2000), 71, 83, 84,
88, 91
Fisher (1936), 115, 143
Fogel *et al.* (1966), 11
Fogel (1995), 1, 11, 12, 15
Glover & Laguna (1997), 13
Golub & Van Loan (1996), 109
Gonzales & Perez (1998), 84, 85
Hall (1998), 102, 155
Härdle (1990), 8, 151
Hartl & Clark (1989), 15
Haykin (1999), 2, 117
Heider & Drabe (1997), 6, 88, 91
Herrera & Magdalena (1998), 82
Hillis (1990), 36
Hoffmann & Nelles (2000), 3, 5, 83, 91, 110
Hoffmann & Pfister (1995), 6
Holland (1975), 12, 15, 19, 37
Homaifar & McCormick (1995), 87, 91
Hornik *et al.* (1989), 4
Ishibuchi *et al.* (2001), 84, 85
Ishibuchi & Nakashima (1999), 6, 47, 84, 88,
91
Iyoda (2000), 20, 29
Iyoda & Von Zuben (2002), 146
Jang (1993), 3–5, 83, 84, 86, 87, 91
Jang *et al.* (1997), 1–3, 47, 55, 57, 63, 66, 68,

- 70, 76, 79
- Jin (2000), 83, 88, 91, 132
- Juillé & Pollack (1996), 139
- Karr (1991b), 3, 83, 87
- Karr (1991a), 87, 91
- Kearns & Vazirani (1994), 151
- Klement *et al.* (2000), 53, 54
- Kosko (1997), 5
- Koza (1992b), 22
- Koza (1992a), 22
- Koza *et al.* (1998), 22
- Lang & Witbrock (1988), 139
- Larsen (1980), 61
- Lee & Takagi (1993), 5, 6, 96
- Lin & Lee (1996), 1–3, 5, 6, 42, 68, 86
- Lukasiewicz (1970), 41, 60
- Mamdani & Assilian (1975), 61, 75, 76
- Michalewicz (1996), 6, 13, 19, 22, 25, 29
- Mitchell (1996), 10, 13, 15, 19, 23, 26
- Moriarty & Miikkulainen (1997), 37, 89
- Moriarty & Miikkulainen (1998), 3, 4, 37, 39,
89, 90
- Moscato (1999), 13
- Nauck & Kruse (1999), 3, 117
- Nomura *et al.* (1991), 85, 91
- Nozaki *et al.* (1997), 83
- Olsson (2001), 35, 36
- Pedrycz & Gomide (1998), 2, 42, 47, 49, 50,
54, 58–60, 65, 66, 71, 73, 81
- Potter & De Jong (2000), 6, 7, 36, 38–40, 89,
90, 140
- Rechenberg (1965), 11
- Rechenberg (1973), 11
- Resher (1969), 41
- Rizzi *et al.* (1999), 5, 59, 86, 117
- Rocha (1992), 2
- Rojas *et al.* (1999), 136
- Rosin & Belew (1997), 35, 36
- Roughgarden (1976), 35
- Sareni & Krähenbühl (1998), 35
- Schwefel (1981), 11
- Setnes *et al.* (1998), 71, 83, 88, 91, 132
- Shi *et al.* (1998), 5, 6
- Shi *et al.* (1999), 6, 91
- Siarry & Guely (1998), 6, 87, 91
- Simpson (1992), 59
- Sober (1975), 153
- Steels (1995), 85
- Takagi & Sugeno (1983), 7, 75, 76
- Takagi & Lee (1993), 5, 7
- Thrift (1991), 87, 91
- Tsukamoto (1979), 76
- Von Zuben (2000), 15, 16, 32
- Vose (1991), 19
- Wineberg (2000), 18
- Wolpert & Macready (1997), 5
- Yager (1988), 54
- Yao (1993), 3, 4
- Yen *et al.* (1998), 88, 110
- Zadeh (1965), 1, 2, 41, 49
- Zadeh (1975), 58, 65
- Hang (1997), 5
- Zimmermann & Zysno (1980), 54

Índice de Assuntos

- abordagem
 - Michigan*, 13, 94
 - Pittsburgh*, 13, 14, 94
- acuidade, 5, 83, 152
- adaptação, 3, 10
 - evolutiva, 17
- agregação, 49
 - das regras, 7, 68, 143
 - dos antecedentes, 3, 7, 62, 103
 - dos conseqüentes, 62
- agrupamento, 59
 - adaptativo, 118
 - nebuloso, 59
- alelos, 10
- alfabeto, 11, 22
- algoritmos
 - co-evolutivos, 36
 - de poda, 111
 - de reparação, 102
 - evolutivos, 13, 14
 - genéticos, 6, 12, 90
 - meméticos, 13
- altura, 47
- ambigüidade, 43
- antecedente, 60
- aplicações práticas, 8
- aprendizado, 8
- aproximação, 107
 - de funções, 83, 121
 - finita, 131
 - universal, 4
- assimetria, 36
- associatividade, 51, 52
- ativação da regra, 73, 76
- auto-organização, 2
- automatização, 5
- autonomia, 85, 153
 - de projeto, 85, 152
- backpropagation, 86, 117
- base de dados, 3, 70
- base de regras, 3, 7, 67, 70
 - específicas, 106
 - genéricas, 106
- busca, 10
 - estocástica, 13
 - local, 2, 13
 - tabu, 13
- caixa-preta, 4
- caixeiro viajante, 20
- camada intermediária, 4, 37
- cardinalidade, 113, 119
- cascata, 88
- centro de

- área, 70, 71
- massa, 70, 71
- ciclo, 119
- classificação de padrões, 139
- classificador nebuloso, 139, 140
- clustering, 86, 154
- clusters, 118
- co-evolução, 6, 35, 38, 90, 152
- co-normas triangulares, 50, 53
- codificação
 - binária, 22
 - inteira, 22
 - mista, 22
 - real, 22
- colunas linearmente
 - dependentes, 110, 113
 - independentes, 111
- combinação
 - convexa, 29
 - linear, 122
- compactação, 5, 85
- competição, 17, 36
- complemento, 49
- completude, 83, 84
- complexidade, 1, 88
- compressão, 59, 118, 149
- computação
 - evolutiva, 1
 - flexível, 1
- comutatividade, 51, 52
- concentração, 48
- condição
 - de parada, 100
 - recíproca, 111, 113
- condições
 - de contorno, 51, 52
 - iniciais, 5, 18
 - irrelevantes, 63
- conexões, 37
- conhecimento do especialista, 3–5, 46, 81
- conjunção, 49, 60
- conjuntos
 - clássicos, 44
 - convexos, 48
 - nebulosos, 2, 41, 44
 - normais, 48
- conseqüentes, 60
 - constantes, 7, 78
 - lineares, 78
 - lingüísticos, 70
 - não-lineares, 7, 78
- consistência, 5, 85
- continuidade, 47
- contra-domínio, 54
- controle, 42
- convergência, 4
- cooperação, 7, 37, 39, 93, 152
- critério de
 - distância, 27, 39
 - erro, 5, 83
- cromossomo, 1, 10
- crossover, 11, 17
 - aritmético, 29, 155
 - de múltiplos pontos, 28, 29
 - simples, 28, 101, 119
 - uniforme, 28, 29, 101

- custo computacional, 8, 125, 153
- dados
 - da IRIS, 115
 - de entrada-saída, 47
 - de teste, 17
 - de treinamento, 17
 - de validação, 121
 - ruidosos, 129
- decodificação, 87, 96
- defuzzyfication, 7, 70
- degeneração, 103
- degradação, 81
- descendentes, 11
- diferenciabilidade, 47
- dilatação, 48
- dispersão, 46, 97
- dispositivos não-paramétricos, 151
- distribuição
 - esparsa, 155
 - normal, 115, 129, 131
 - uniforme, 25
- diversidade, 24
- domínio, 29–31, 54, 58, 70
- domínios de aplicação, 8, 38, 115
- don't care conditions, 47, 63, 84
- ecossistema, 1, 38
- elitismo, 26
- epigênese, 16
- epocas, 119
- erro de
 - classificação, 144
 - aproximação, 115, 126
 - previsão, 130
 - teste, 123
 - treinamento, 121
- espécie, 1, 11
- espaço
 - fenotípico, 15
 - genotípico, 15
 - multimodal, 13
- especialista, 42, 59
- esperança matemática, 130
- espirais concêntricas, 115, 139
- estratégias evolutivas, 11–13
- evolução, 1, 123
 - hierárquica, 90
 - natural, 10–12, 15
 - tradicional, 126, 139
- experimento de caminho fechado, 116, 146
- exploração, 2, 12, 13, 25
- exploração, 2, 13
- extensão cilíndrica, 65
- factibilidade, 23, 29
- fenótipo, 15
- fitness, 7, 17, 21
- flexibilidade, 129, 153
- função de
 - desempenho, 14, 20, 21
 - irrelevância, 47, 63, 97
 - pertinência, 2, 44, 93
- funções quadráticas, 120, 134
- fuzzy clusters, 59
- genótipo, 15
- gene, 1, 10, 15, 22

- generalização, 2, 17
- genoma, 11, 15
- geração, 2, 16
- grânulo, 67
- grafo, 56
- grafo nebuloso, 68
- gramática, 58
- granularidade, 59
 - máxima, 99
- granularização, 59, 93
- heurística, 20, 89
- hierarquia, 7, 94, 152
- hospedeiro-parasita, 36
- hospedeiros, 36
- idempotência, 53
- implicação
 - de Lukasiewicz, 61
 - lógica, 41, 60
 - nebulosa, 60, 61
- imprecisão, 1, 43
- incerteza, 43
- inferência
 - composicional, 71
 - escalonada, 73
- inovação, 10
- instalações industriais, 42
- integridade lingüística, 88
- inteligência computacional, 151
- intensificação de contraste, 49
- interação, 105
- interdependência, 36, 93
- interpretabilidade, 83, 152
- interseção, 49
- iteração, 13, 14, 17, 119
- lógica
 - bi-valores, 41
 - clássica, 61
 - multi-valores, 41
 - nebulosa, 41
- lei
 - da contradição, 50, 52
 - da exclusão do meio, 50, 52
 - de Morgan, 50
- linguagens de programação, 156
- lista de atributos, 11, 19
- máquinas
 - de estado finito, 11, 15
 - inteligentes, 1
- média
 - dos máximos, 70
 - generalizada, 54
 - ponderada, 75
 - ordenada, 54
- método
 - de otimização global, 7, 107
 - de otimização local, 7, 110
 - dos quadrados mínimos, 88
 - ponderados, 7, 110
 - recursivo, 86
 - do gradiente, 6
 - forte, 19
 - fraco, 19
 - hill climbing, 12
 - simulated annealing, 13

- módulos, 38, 90, 101
- mínimos locais, 6, 87
- mapeamento entrada-saída, 106, 107
- matching, 73–76
- matlab, 156
- matriz de posto completo, 110
- mecanismo de raciocínio, 3
- migração, 16, 17
- modelo
 - Mamdani, 7, 76
 - matemático, 42
 - não-paramétrico, 152
 - Takagi-Sugeno, 7, 76
- modificadores, 58
- modularidade, 6
- modus ponens, 63
 - generalizado, 63, 64
- monotonicidade, 51
- mutação, 11, 17, 29
 - inteira, 30
 - real, 30
 - não-uniforme, 30
 - uniforme, 30
 - simples, 30
- núcleo, 47, 48
- número de condição, 111
- nível de similaridade, 73
- não-linearidade, 2, 109
- necessidade, 54
- neurônios, 2, 37
- nicho, 35, 38
- normalização, 48
- normas triangulares, 50, 52, 53
- norma Euclidiana, 109, 111
- operadores
 - compensatórios, 54
 - evolutivos, 36, 100
 - genéticos, 13, 126
- operador humano, 42
- ordenação, 25, 29, 36
- organismos diplóides, 11, 12
- overfitting, 129
- pétala, 143
- parâmetros de sistemas nebulosos, 4
- paralelismo, 6
- parasitas, 36
- parcimônia, 153
- pares
 - de crossover, 27, 100
 - ordenados, 44
- partição nebulosa, 7, 59, 90
 - esparsa, 59
 - fina, 59
 - uniforme, 59
- penalidade, 23
- perturbações aleatórias, 129
- plausibilidade, 43
- pleiotropia, 15, 27
- poligenia, 15, 27
- ponto de
 - crossover, 28
 - mutação, 29
- população, 7, 13
- possibilidade, 43, 54
- pressão seletiva, 23–25

- previsão de séries temporais, 68
- processamento de informação, 2
- processo de poda, 109, 111
- produto
 - algébrico, 56
 - cartesiano, 54
- produtos manufaturados, 42
- programação
 - evolutiva, 11
 - genética, 22
 - multi-objetivo, 155
- projeção, 65
- projetista, 3
- projeto automático, 3, 81
- proposições, 47, 60
 - atômicas, 60, 73
 - monovariáveis, 61
 - multivariáveis, 62
 - nebulosas, 41
- rótulo, 58
- raciocínio
 - aproximado, 42, 63
 - humano, 42, 63
 - nebuloso, 3, 64
- recombinação, 11
- reconhecimento de padrões, 42
- redes neurais, 2, 126
- redundância, 7
- regiões promissoras, 12, 18
- regra
 - composicional de inferência, 42, 64
- regras
 - individuais, 7, 93
 - lingüísticas, 81, 84
 - nebulosas, 2, 60
- relações nebulosas, 54
- representação
 - cromossômica, 6, 82
 - distribuída, 2
 - do conhecimento, 3, 42, 81
- reprodução, 12
- restrições ambientais, 18
- robótica, 68
- robustez, 2
- ruído branco, 115, 129
- sépala, 143
- símbolos ambientais, 17
- saída
 - crisp, 70, 71
 - desejada, 155
 - nebulosa, 79
- seleção, 12, 100
 - aleatória, 26
 - de variáveis, 102, 153
 - natural, 9
 - por
 - classificação, 25
 - desempenho, 13
 - diversidade, 26, 100, 119
 - roleta, 24
 - torneio, 25
 - steady-state, 26
- semântica da regra, 7, 60, 93
- simplicidade, 84
- singleton, 45
- sistemas

- adaptativos, 9
- co-evolutivos, 116, 141
- computacionais, 3, 15
- dinâmicos, 155
- evolutivos, 9, 11, 15
- híbridos, 1, 4, 5, 117
- inteligentes, 68, 81
- nebulosos, 2, 42, 68, 99
- neuro-nebulosos, 4
- sistemas classificador, 37
- sobre-ajuste, 129
- sobreposição, 84
- sobrevivência genotípica, 17
- soluções
 - ótimas, 32
 - candidatas, 11
 - desejadas, 12
 - factíveis, 23
 - infactíveis, 23
 - parciais, 7, 90
 - parcimoniosas, 151
 - potenciais, 3
- soma simétrica, 54
- suavidade, 20, 135
- subcomponentes, 6
- subconjunto nebuloso, 48
- subjetividade, 42
- super-indivíduo, 23, 25
- superfície de desempenho, 17
- suporte, 3, 47, 102, 133
- taxa de
 - crossover, 27
 - mutação, 29
- tecnologia da informação, 42
- teoria da probabilidade, 43
- termos
 - lingüísticos, 2, 58
 - não-primários, 58
 - primários, 58
 - quadráticos, 120
 - redundantes, 113
- tomada de decisão, 6, 68
- transparência, 83, 88
- tratabilidade computacional, 4, 153
- união, 49
- uniformidade, 96, 99
- universo, 44
 - de discurso, 58
- usuário, 3
- valor
 - absoluto, 97
 - modal, 46, 97
 - verdade, 41
- variável
 - aleatória, 25, 130
 - base, 106, 128
 - de entrada, 102
 - de saída, 87, 153
 - irrelevante, 85, 102, 106
- variância do erro, 130–132
- variação
 - aleatória, 9, 30
 - genética, 10
- verossimilhança, 43
- vida artificial, 42

visão computacional, 42

visibilidade, 5, 83