

Universidade Federal de São Carlos

Centro de Ciências Exatas e de Tecnologia

Programa de Pós-Graduação em Ciência da Computação

**Abordagem Co-Evolutiva Hierárquica para Geração
Automática de Sistemas Nebulosos**

Anderson Francisco Talon

São Carlos
Setembro de 2006

**Ficha catalográfica elaborada pelo DePT da
Biblioteca Comunitária da UFSCar**

T152ac

Talon, Anderson Francisco.
Abordagem co-evolutiva hierárquica para geração
automática de sistemas nebulosos / Anderson Francisco
Talon -- São Carlos : UFSCar, 2007.
102 f.

Dissertação (Mestrado) -- Universidade Federal de São
Carlos, 2006.

1. Inteligência artificial. 2. Coevolução. 3. Algoritmos
genéticos. 4. Sistema Fuzzy-genético. 5. Sistema genético
nebuloso. I. Título.

CDD: 006.3 (20^a)

Universidade Federal de São Carlos

Centro de Ciências Exatas e de Tecnologia

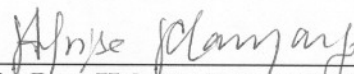
Programa de Pós-Graduação em Ciência da Computação

“Abordagem Co-Evolutiva Hierárquica para Geração Automática de Sistemas Nebulosos”

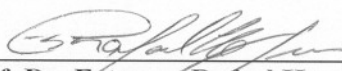
ANDERSON FRANCISCO TALON

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Membros da Banca:



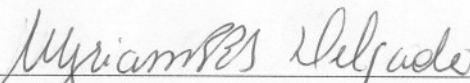
Profa. Dra. Heloisa de Arruda Camargo
(Orientadora – DC/UFSCar)



Prof. Dr. Estevam Rafael Hruschka Junior
(DC/UFSCar)



Profa. Dra. Maria do Carmo Nicoletti
(DC/UFSCar)



Profa. Dra. Myriam Regattieri De Biase da Silva
Delgado
(Universidade Tecnológica Federal do Paraná –
Dainf)

São Carlos
Setembro/2006

AGRADECIMENTOS

A minha família pelo apoio e incentivo.

A minha orientadora Professora Doutora Heloisa de Arruda Camargo.

Aos membros da banca examinadora.

E a todos que contribuíram direta ou indiretamente nesse trabalho.

RESUMO

Este trabalho enfoca o problema de geração automática de sistemas nebulosos por meio da computação evolutiva, mais especificamente por meio da abordagem de co-evolução. A co-evolução baseia-se na idéia de modelagem modular de subcomponentes do problema. Neste trabalho esses subcomponentes são representados por espécies diferentes, que têm uma relação colaborativa entre si. Essa relação força uma evolução co-adaptada entre as populações das espécies. O sistema nebuloso a ser gerado tem como objetivo a classificação de padrões. Basicamente, o esquema evolutivo é formado por quatro espécies diferentes, que têm uma colaboração hierárquica, tanto na geração das espécies, quanto na determinação da aptidão dos indivíduos dessas espécies. Essas espécies são organizadas em níveis, onde a colaboração na geração das espécies se dá dos níveis mais baixos para os mais altos, e a colaboração na determinação das aptidões se dá dos níveis mais altos para os mais baixos. Os resultados obtidos indicam que a abordagem estudada é bastante promissora e pode-se obter, por meio dela, sistemas de classificação eficientes que apresentam desempenho semelhante ao de outras abordagens encontradas na literatura.

Palavras-Chaves: Co-Evolução, Classificador Nebuloso, Algoritmos Genéticos, Sistemas Nebulosos, Sistemas Nebulosos Genéticos, Geração Automática.

ABSTRACT

This work focuses on the problem of automatic generation of fuzzy systems through evolutionary computation, specifically using the approach of co-evolution. Co-evolution is based on the idea of modular modeling of the problem subcomponents. In this work the subcomponents are represented by different species, which have a collaborative relation among them. The fuzzy system to be created has the objective of pattern classification. Basically, the evolutionary scheme is composed by four different species, which have a hierarchical collaboration both in the generation of the species and in the fitness determination of the individuals of these species. These species are organized in levels, where the contribution in the species generation happens from the lowest to highest levels and the contribution in the fitness determination happens from the highest to lowest levels. The results obtained indicate that the studied approach is very promising and, through its use, one can generate efficient classification systems, that present a performance similar to other approaches found in the literature.

Keywords: Co-Evolution, Fuzzy Classifier, Genetic Algorithms, Fuzzy Systems, Genetic Fuzzy Systems, Automatic Generation.

ÍNDICE

1.Introdução.....	1
1.1. Objetivo	3
1.2. Organização do Trabalho.....	4
2.Sistemas Nebulosos	5
2.1. Conjuntos Nebulosos	6
2.1.1. Tipos de Funções de Pertinência	8
2.1.2. Definições Básicas.....	9
2.2. Operações em Conjuntos Nebulosos	10
2.2.1. Operações Simples (argumento único).....	11
2.2.2. Operações Compostas (múltiplos argumentos).....	11
2.3. Relações Nebulosas	14
2.4. Computação com Regras Nebulosas	15
2.4.1. Variáveis Lingüísticas	16
2.4.2. Sintaxe das Regras Nebulosas	17
2.4.3. Semântica das Regras Nebulosas	19
2.4.4. Inferência com Regras Nebulosas	21
2.5. Sistemas Nebulosos Baseados em Regras	22
2.5.1. Sistemas Nebulosos de Classificação	23
3.Computação Evolutiva	25
3.1. Técnicas de Computação Evolutiva	27
3.2. Algoritmos Genéticos	28
3.2.1. Codificação.....	31
3.2.2. Métodos de Seleção	32
3.2.3. Operadores Genéticos.....	35
3.3. Co-Evolução	38
3.3.1. Abordagens Co-Evolutivas.....	41
4.Sistemas Nebulosos Genéticos	45
4.1. Abordagens de Sistemas Nebulosos Genéticos	48
4.1.1. Adaptação Genética	49
4.1.2. Construção Genética.....	50
4.1.3. Abordagens Baseadas em Co-Evolução.....	55

5.Sistema de Classificação Nebuloso Co-Evolutivo	57
5.1. Abordagem Co-Evolutiva do Classificador Nebuloso	58
5.2. Codificação Genética.....	60
5.3. Relações de Colaboração.....	66
5.4. Cálculo das Aptidões	67
5.5. População Inicial	69
5.6. Método de Seleção.....	70
5.7. Operadores Evolutivos	70
5.8. Algoritmo Genético	71
6.Simulações e Resultados	72
6.1. Abordagens de Comparação.....	73
6.2. Considerações Preliminares.....	73
6.3. Espirais Concêntricas	75
6.4. IRIS.....	79
6.5. AUTO-MPG	82
7.Considerações Finais	84
7.1. Trabalhos Futuros	85
Apêndice A.....	87
BIBLIOGRAFIA	89

Capítulo 1

Introdução

A Computação Flexível (*Soft Computing*) é um paradigma abrangente que agrega tradicionalmente as metodologias de redes neurais artificiais, dos sistemas nebulosos e da computação evolutiva. A evolução observada nesse campo nos últimos anos reforça a tendência cada vez mais evidente de tratar seus componentes como metodologias complementares, que dispõem de recursos adequados ao desenvolvimento de sistemas híbridos com as mais diversas finalidades. As potencialidades dos componentes envolvidos e as suas possibilidades de combinação oferecem uma nova visão para resolução de problemas complexos que muitas vezes não poderiam ser atacados de outra forma, particularmente no que diz respeito ao tratamento da incerteza e imprecisão (Pedrycz, 1998).

Uma importante e bem sucedida forma de hibridação no contexto da computação flexível refere-se à geração automática de sistemas nebulosos pela exploração da capacidade de aprendizado e generalização da computação evolutiva e das redes neurais.

Os Sistemas Nebulosos incorporam metodologias e técnicas fundamentadas nos conceitos de conjuntos nebulosos e lógica nebulosa (Zadeh, 1965; 1973) para representar e processar informações lingüísticas, e fornecerem mecanismos para tratar a imprecisão e a incerteza inerentes ao conhecimento do mundo real. Por isso 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 (Pedrycz, 1996; Pedrycz & Gomide 1998; Dumitrescu *et al.*,

2000). Um tipo específico de Sistemas Nebulosos de interesse particular para este trabalho são os Sistemas Nebulosos Baseados em Regras, que utilizam regras *se-então* para representar o conhecimento. As regras são uma maneira conveniente e popular para expressar o conhecimento, proporcionando transparência e compreensibilidade ao sistema.

Um Sistema Nebuloso Baseado em Regras é composto por dois componentes principais: a Base de Conhecimento e o Mecanismo de Inferência.

A Base de Conhecimento é composta pela Base de Regras, que armazena o conjunto de regras nebulosas que modelam conhecimento sobre um determinado problema e pela Base de Dados, que contém as definições dos conjuntos nebulosos relacionados aos termos lingüísticos usados nas regras nebulosas. O Mecanismo de Inferência é responsável pelo processamento das regras, o qual é realizado por um método de raciocínio nebuloso.

Uma das tarefas mais importantes e difíceis no desenvolvimento de um Sistema Nebuloso Baseado em Regras é gerar a sua Base de Conhecimento. Nos últimos anos, métodos automáticos têm sido utilizados para geração das Bases de Conhecimento, que consistem em extrair automaticamente o conhecimento necessário a partir de dados numéricos que representam amostras ou exemplos do problema.

Apesar de sua reconhecida e comprovada eficácia, os Sistemas Nebulosos não dispõem de mecanismo próprio para aprendizado de seus componentes, o que tem motivado a procura por abordagens híbridas que explorem a capacidade de aprendizado de outras metodologias de forma combinada, para gerar automaticamente Sistemas Nebulosos completos ou partes deles.

Dentre as iniciativas mais bem sucedidas na busca por esse objetivo estão aquelas que usam ferramentas de aprendizado baseadas em algoritmos de agrupamento (*clustering*) (Liao *et al.*, 1997), métodos baseados em gradiente (Nomura *et al.*, 1992), redes neurais (Nauck & Kruse, 1997) e algoritmos genéticos (Cordón *et al.*, 2001a), sendo esses dois últimos investigados no contexto da computação flexível.

O trabalho aqui apresentado enfoca a geração automática de Sistemas Nebulosos de classificação por meio de Algoritmos Genéticos. De uma maneira geral, cunhou-se o termo Sistemas Nebulosos Genéticos (SNG) para nomear um Sistema Nebuloso combinado com um processo de aprendizado baseado em Algoritmos Genéticos. Uma coleção significativa e abrangente de modalidades e de novas tendências na área de Sistemas Nebulosos Genéticos pode ser encontrada em (Cordón *et al.* 2004b).

Os Algoritmos Genéticos estão sendo utilizados com sucesso sob diferentes enfoques no que diz respeito à geração da Base de Conhecimento de Sistemas Nebulosos. Em (Yuan & Zhuang, 1996), (Gonzalez & Perez, 1999) e (Ishibuchi *et al.*, 1999b) é usado um AG para gerar somente a base de regras nebulosas para problemas de classificação de padrões. A definição genética de conjuntos nebulosos para problemas de controle pode ser encontrada em (Karr, 1991b) e (Cordón *et al.*, 2001b,c). A base de conhecimento completa (regras e conjuntos nebulosos) foi gerada usando algoritmos genéticos em (Homaifar & McCormick, 1995), (Magdalena & Velasco, 1996; 1997) e (Cordón & Herrera, 2001). O ajuste dos parâmetros das funções de pertinência dos conjuntos nebulosos é apresentado em (Herrera *et al.*, 1995) e (Gudwin *et al.*, 1998). A otimização da base de regras, a fim de eliminar regras redundantes e desnecessárias, pode ser encontrada em (Ishibuchi *et al.*, 1997). Há ainda trabalhos que geram um sistema nebuloso completo, incluindo a base de regras, a base de dados e operadores que definem o mecanismo de inferência (Delgado *et al.*, 2002; 2004).

Qualquer uma das tarefas previamente citadas pode ser considerada como um processo de busca no espaço das possíveis soluções. Por isso, a combinação de Sistemas Nebulosos e Algoritmos Genéticos tem grande aceitação na comunidade científica, uma vez que estes algoritmos são robustos e possuem a capacidade de percorrer de forma global espaços de busca irregulares e extensos, encontrando uma solução ótima ou quase ótima rapidamente (Yuan & Zhuang, 1996).

Este trabalho dá continuidade à investigação que vem sendo desenvolvida no contexto de um projeto mais abrangente intitulado *Construção e Otimização de Base de Conhecimento Fuzzy (Uma Abordagem Evolutiva)*, que estuda abordagens envolvendo o uso de algoritmos genéticos na geração de sistemas nebulosos. Os principais resultados dos trabalhos estão relatados em (Castro *et al.*, 2003; 2004a,b,c,d; Castro & Camargo, 2004; 2005a,b; Pires & Camargo, 2004).

1.1. Objetivo

O objetivo deste trabalho é investigar a geração automática de um Sistema Nebuloso completo pela abordagem co-evolutiva, desde as funções de pertinência que definem os conjuntos nebulosos e as regras nebulosas, até o mecanismo de inferência.

Como o problema de geração automática de Sistemas Nebulosos é complexo, a utilização do Algoritmo Genético tradicional pode ser inviável devido ao tamanho do problema. A utilização da abordagem co-evolutiva tem vantagem sobre o Algoritmo Genético tradicional por dividir um problema grande em subproblemas menores, onde todos os subproblemas contribuem para a solução do problema maior.

O sistema proposto é baseado no trabalho desenvolvido por Delgado (2002). Entretanto, diferente da proposta original, este trabalho visa a geração de um classificador nebuloso. No caso do modelo proposto por Delgado (2002), o modelo gerado é do tipo Takagi-Sugeno e foca principalmente o problema de aproximação de funções, embora esse modelo tenha sido aplicado em alguns problemas de classificação na etapa de simulação. Já o modelo desenvolvido neste trabalho é de um classificador nebuloso voltado unicamente para a tarefa de classificação. Além da questão do foco do trabalho, outra diferença entre a proposta apresentada aqui e o trabalho que a inspirou diz respeito às mudanças feitas nas funções do cálculo das aptidões. Tais mudanças visam a melhoria da interpretabilidade, levando em conta a compactação da base de regras e a simplicidade das regras nebulosas no Sistema Nebuloso.

Espera-se que os Sistemas Nebulosos gerados apresentem acuidade e interpretabilidade. A acuidade está relacionada ao desempenho em termos de erro. A interpretabilidade está na facilidade com que a solução encontrada é compreendida e interpretada. A interpretabilidade consiste na visibilidade da partição nebulosa, na simplicidade das regras, na compactação da base de regras, e na consistência de base de regras, ou seja, a não existência de regras conflitantes.

1.2. Organização do Trabalho

Esta dissertação está dividida em capítulos, organizados da seguinte forma: no Capítulo 2 são apresentados os Sistemas Nebulosos; no Capítulo 3 são introduzidas as técnicas de computação evolutiva, especificamente os Algoritmos Genéticos e a co-evolução; no Capítulo 4 são descritos os Sistemas Nebulosos Genéticos; no Capítulo 5 é apresentado o trabalho desenvolvido; no Capítulo 6 são mostrados as simulações e os resultados do sistema; e por fim, no Capítulo 7, são apresentadas as considerações finais desse trabalho.

Capítulo 2

Sistemas Nebulosos

A designação de um sistema como Sistema Nebuloso é bastante vaga e genérica, e diversas caracterizações podem ser encontradas na literatura. De maneira geral, podemos dizer que Sistemas Nebulosos são sistemas que incorporam mecanismos baseados na teoria de conjuntos nebulosos e lógica nebulosa para tratamento da imprecisão por meio de variáveis cujos valores são expressos por conjuntos nebulosos – as variáveis lingüísticas. Uma característica central a esses sistemas, que está presente em todas as tentativas de definição do termo, é a de que eles se baseiam no conceito de codificação da informação por meio de conjuntos nebulosos, o que permite a generalização da informação, associada à sua imprecisão (Yager & Filev, 1994).

Os seres humanos são capazes de lidar com processos bastante complexos, com base em informações imprecisas ou aproximadas. Por outro lado, técnicas tradicionais quantitativas exigidas para o tratamento computacional de diversas classes de problemas dificultam a representação da complexidade do mundo real, para a qual o ser humano tem habilidades natas. O extenso repertório de técnicas para tratamento de imprecisão derivadas a partir da teoria de conjuntos nebulosos proposta por Zadeh (1965) surgiu como uma forma de flexibilizar os modelos tradicionais rígidos e precisos e permitir o tratamento de situações que são naturalmente imprecisas. O resultado dessa nova abordagem deu origem a uma nova classe de sistemas aos quais nos referimos aqui por Sistemas Nebulosos (SN) - também chamados de sistemas difusos ou de sistemas *fuzzy*.

O sucesso dos Sistemas Nebulosos é cada vez mais evidente devido ao crescente volume de trabalhos das últimas décadas, abordando tanto seus aspectos teóricos como

aplicações práticas. Noções essenciais quanto à teoria, questões de modelagem dos sistemas e aplicações podem ser obtidas, por exemplo em Klir & Yuan (1995), Pedrycz (1996) e Pedrycz & Gomide (1998).

Dentro dessa vasta classe, interessa-nos um tipo particular de sistemas nebulosos, chamados de Sistemas Nebulosos Baseados em Regras (SNBR), que são sistemas compostos de um conjunto de regras *se-então* que modelam o conhecimento de um determinado domínio com a finalidade de resolver algum problema, usando variáveis com valores lingüísticos. Essas regras são acompanhadas de conjuntos nebulosos e associadas a um mecanismo de inferência que realiza o processamento das regras.

Este capítulo descreve os conceitos básicos relativos a conjuntos nebulosos e lógica nebulosa necessários à compreensão deste trabalho, além do método de inferência de interesse específico para a abordagem de construção de regras nebulosas apresentada aqui, relacionado com o problema de classificação nebulosa.

2.1. Conjuntos Nebulosos

A noção básica de conjunto como uma coleção de objetos está presente em todos os aspectos da nossa vida diária. Utilizamos a noção de conjunto sempre que precisamos organizar, resumir e generalizar o conhecimento sobre objetos.

Na teoria clássica dos conjuntos, lidamos com uma dicotomia: um objeto pertence ou não pertence ao conjunto. Um objeto pertencente a um conjunto é dito um elemento do conjunto. Escrevemos $x \in A$ para indicar que o objeto x pertence ao conjunto A e escrevemos $x \notin A$ para indicar que o objeto x não pertence ao conjunto A . Uma das maneiras de se definir um conjunto é por meio de uma *função característica*, que associa o valor 1 (um) ao elemento se ele pertence ao conjunto ou 0 (zero) se ele não pertence ao conjunto. Por exemplo, um conjunto A sobre um universo \mathbf{X} pode ser definido por uma função $A: \mathbf{X} \rightarrow \{0,1\}$ como:

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

Assim, existe uma distinção clara e bem determinada entre os objetos que estão e os que não estão no conjunto.

Entretanto, o mundo real apresenta situações em que os conjuntos clássicos não são flexíveis o suficiente para uma descrição apurada de significado, tendo em vista que eles forçam de forma abrupta a transição de pertinência absoluta para a não pertinência absoluta.

Os conjuntos nebulosos, introduzidos por Zadeh (1965), são conjuntos cujos limites não são precisos, permitindo a pertinência parcial de cada elemento ao conjunto. Sendo A um conjunto nebuloso, um objeto x pode pertencer ao conjunto A com um grau, que é chamado de *grau de pertinência* de x em A e usualmente expresso por um número no intervalo $[0,1]$. Os valores extremos desse intervalo, 0 (zero) e 1 (um), representam, a não pertinência total ao conjunto e a pertinência total ao conjunto respectivamente. Assim, a teoria de conjuntos clássicos pode ser considerada um caso restrito da teoria dos conjuntos nebulosos, em que somente os dois graus extremos de pertinência são permitidos. Uma das principais utilidades da transição gradual entre pertinência e não pertinência é permitir a representação de conceitos imprecisos expressos em linguagem natural, que não têm uma distinção rigorosa entre membros e não membros. Como exemplo, podemos citar os conceitos de pessoa alta, imóvel caro, temperatura baixa, tráfego intenso e outros.

Os conjuntos nebulosos são caracterizados formalmente por uma *função de pertinência*, que é uma generalização da função característica dos conjuntos clássicos. A função de pertinência associa, a cada elemento do universo de discurso, um valor dentro de um intervalo específico, que indica o grau de pertinência do elemento ao conjunto. O intervalo de valores mais utilizado é o intervalo $[0,1]$. Indicamos que a função de pertinência de um conjunto A mapeia elementos do universo \mathbf{X} em graus de pertinência no intervalo $[0,1]$ escrevendo $A: \mathbf{X} \rightarrow [0,1]$ (Klir & Yuan, 1995).

Um conjunto nebuloso A sobre o universo \mathbf{X} , pode ser representado por um conjunto de pares ordenados tais que o segundo elemento do par é um objeto $x \in \mathbf{X}$, e o primeiro elemento do par é seu respectivo grau de pertinência no conjunto nebuloso A , ou seja:

$$A = \{A(x)/x \mid x \in \mathbf{X}\}$$

Para distinguir entre conjuntos nebulosos e conjuntos clássicos, muitas vezes fazemos referência aos conjuntos clássicos usando o termo do inglês *crisp*.

2.1.1. Tipos de Funções de Pertinência

Em geral, conjuntos nebulosos são definidos para representar conceitos imprecisos, o que depende do entendimento que se tem desse conceito e também do contexto em que será utilizado. Apesar de teoricamente qualquer função da forma $A: X \rightarrow [0,1]$ descrever uma função de pertinência associada a um conjunto nebuloso A , a representação de conceitos como conjuntos nebulosos no contexto de uma aplicação pode exigir que essas funções possuam certas formas específicas ou até mesmo certas propriedades. A rigor, a forma mais adequada de uma função de pertinência só pode ser definida no contexto de uma aplicação, mas algumas formas de funções de pertinência tornaram-se mais conhecidas e são utilizadas frequentemente, pela sua simplicidade. Entre as formas de funções mais conhecidas estão as famílias de funções parametrizadas, sendo que algumas delas são apresentadas a seguir:

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

$$A(x) = \begin{cases} 0 & \text{se } x \leq a \\ \frac{x-a}{m-a} & \text{se } x \in (a, m) \\ 1 & \text{se } x = m \\ \frac{b-x}{b-m} & \text{se } x \in (m, b) \\ 0 & \text{se } x \geq b \end{cases}$$

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

$$A(x) = \begin{cases} 0 & \text{se } x \leq a \\ \frac{x-a}{m-a} & \text{se } x \in (a, m) \\ 1 & \text{se } x \in [m, n] \\ \frac{b-x}{b-n} & \text{se } x \in (n, b) \\ 0 & \text{se } x \geq b \end{cases}$$

- Função Gaussiana: parâmetros (m, k) , onde $k > 0$.

$$A(x) = e^{-k(x-m)^2}$$

A figura 2.1 ilustra os formatos descritos antes, especificando os parâmetros associados a cada caso.

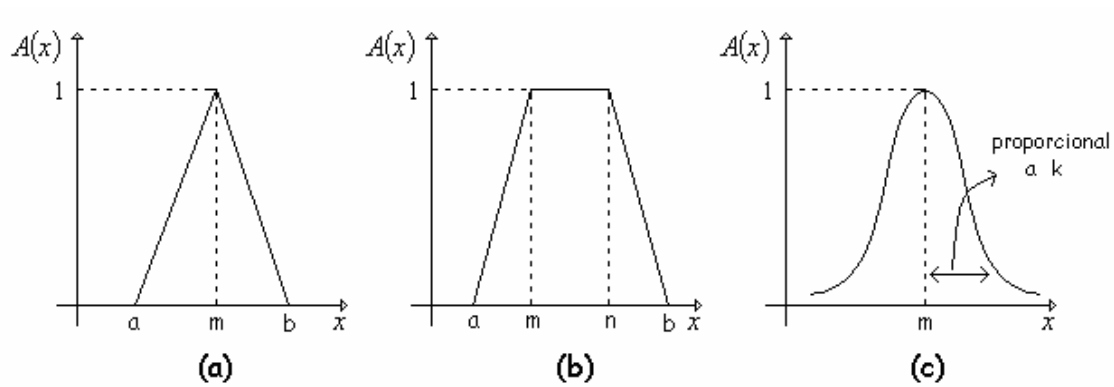


Figura 2.1: Exemplos de funções de pertinência: (a) triangular, (b) trapezoidal e (c) gaussiana.

Um tipo bastante específico de conjunto nebuloso que pode ter utilidade em aplicações é o chamado conjunto nebuloso unitário, ou *fuzzy singleton*. Nesse conjunto, existe apenas um ponto x no universo \mathbf{X} para o qual o valor da função de pertinência é não-nulo.

2.1.2. Definições Básicas

Um conjunto nebuloso pode ser caracterizado por alguns atributos adicionais, entre os quais aqueles que estão associados aos conceitos de suporte, núcleo, altura e normalidade, apresentados a seguir:

- Suporte de um conjunto nebuloso A (S_A): conjunto clássico (*crisp*) que contém todos os elementos do conjunto universo \mathbf{X} que têm grau de pertinência diferente de 0 (zero) em A .

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

- Núcleo de um conjunto nebuloso A (N_A): conjunto clássico (*crisp*) que contém todos os elementos do conjunto universo \mathbf{X} que têm grau de pertinência igual a 1 (um) em A .

$$N_A = \{x \mid A(x) = 1\}$$

- Altura de um conjunto nebuloso A (H_A): maior grau de pertinência obtido por qualquer elemento do conjunto A (sup denota supremo).

$$H_A = \sup_x \{A(x)\}$$

- Conjunto Normal: um conjunto nebuloso é dito normal se sua altura for igual a 1 (um). Caso contrário é chamado de subnormal. O núcleo de um conjunto subnormal é um conjunto vazio.

Um dos conceitos mais importantes relacionados a conjuntos nebulosos é o conceito de Corte- α e sua variante Corte- α Forte, definidos a seguir:

- Corte- α : conjunto clássico (*crisp*) que contém todos os elementos do conjunto universo X que têm grau de pertinência maior ou igual que α , para algum $\alpha \in [0,1]$.

$$A_{\alpha} = \{x \mid A(x) \geq \alpha\}$$

- Corte- α Forte: conjunto clássico (*crisp*) que contém os elementos do conjunto universo X que têm grau de pertinência maior que α , para algum $\alpha \in [0,1]$.

$$A_{\alpha+} = \{x \mid A(x) > \alpha\}$$

Associado ao conceito de corte- α definimos o Conjunto de Níveis de A , como o conjunto de todos os valores $\alpha \in [0,1]$ que representam cortes- α distintos de um dado conjunto nebuloso A , ou seja:

$$\eta_A = \{\alpha \mid A(x) = \alpha, \forall x \in X\}$$

O conceito de corte- α é utilizado na construção do *Teorema da Representação* que estabelece que todo conjunto nebuloso A pode ser decomposto em um conjunto de seus cortes- α . A apresentação desse teorema não foi incluída aqui por não ser de interesse deste trabalho. Para mais detalhes, ver (Klir & Yuan, 1995 e Pedrycz & Gomide, 1998).

2.2. Operações em Conjuntos Nebulosos

As operações sobre conjuntos nebulosos são essenciais no processo de representação e processamento de conhecimento na forma de conjuntos nebulosos. Discutiremos as operações sobre conjuntos nebulosos distinguindo entre os casos das operações de argumento único, que modificam o formato da função de pertinência que define o conjunto nebuloso, e das operações de múltiplos argumentos, que relacionam dois ou mais conjuntos nebulosos. Essas operações serão apresentadas a seguir.

2.2.1. Operações Simples (argumento único)

Essas operações possuem um argumento único, e elas modificam o formato da função de pertinência. A seguir, serão apresentadas essas operações.

- Normalização: transforma um conjunto nebuloso subnormal A em um conjunto nebuloso normal A^N por:

$$A^N(x) = A(x) / H_A$$

sendo H_A a altura de A .

- Concentração: define um conjunto nebuloso A^C tornando os valores de pertinência de A relativamente menores.

$$A^C(x) = A^2(x)$$

- Generalização da Concentração.

$$A^C(x) = A^p(x), p > 1$$

- Dilatação: define um conjunto nebuloso A^D tornando os valores de pertinência de A relativamente maiores.

$$A^D(x) = A^{0.5}(x)$$

- Generalização da Dilatação.

$$A^D(x) = A^p(x), p \in (0;1)$$

- Intensificação de contraste: resulta no conjunto A^I combinando as operações de dilatação e concentração. Os valores de pertinência menores que $\frac{1}{2}$ diminuem e os maiores aumentam.

$$A^I(x) = \begin{cases} 2 * A^2(x) & \text{se } A(x) \in [0;0,5] \\ 1 - 2 * (1 - A(x))^2 & \text{se } A(x) \in (0,5;1] \end{cases}$$

2.2.2. Operações Compostas (múltiplos argumentos)

As operações conhecidas da teoria de conjuntos clássicos – união, interseção e complemento – podem ser generalizadas para o contexto dos conjuntos nebulosos. Essa generalização pode ser feita de formas diferentes, mas uma delas em particular tem um significado especial na teoria de conjuntos nebulosos, por isso as operações assim

obtidas são chamadas de operações padrão. Apresentamos a seguir as operações padrão e, na seqüência, as operações generalizadas.

Operações Padrão

As operações sobre conjuntos nebulosos referidas como operações padrão têm uma importância destacada na teoria de conjuntos nebulosos por possuírem certas propriedades que dá a elas um significado especial (Klir & Yuan, 1995). Na seqüência são apresentadas as definições de tais operações.

Sejam A e B dois conjuntos nebulosos no universo X com suas funções de pertinência $A(x)$ e $B(x)$, respectivamente.

A interseção nebulosa padrão ($A \cap B$) dos conjuntos A e B é definida por:

$$(A \cap B)(x) = \min[A(x), B(x)] = A(x) \wedge B(x)$$

onde \min e \wedge denotam o operador de mínimo.

A união nebulosa padrão ($A \cup B$) dos conjuntos A e B é definida por:

$$(A \cup B)(x) = \max[A(x), B(x)] = A(x) \vee B(x)$$

onde \max e \vee denotam o operador de máximo.

O complemento nebuloso de A indica o grau com que x não pertence ao conjunto A , e pode ser definido por:

$$\overline{A}(x) = 1 - A(x)$$

A figura 2.2 mostra exemplos de união, interseção e complemento padrão aplicados a dois conjuntos nebulosos A e B .

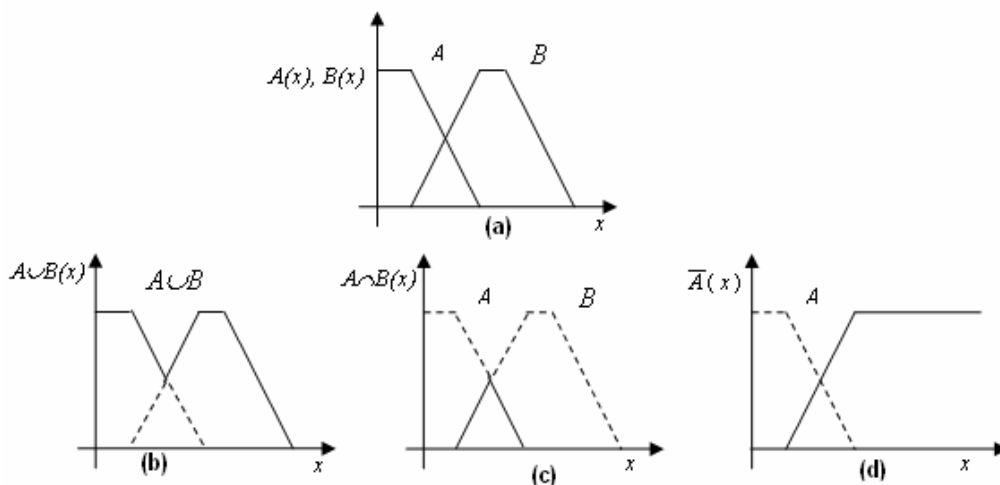


Figura 2.2: (a) Conjuntos A e B, (b) União, (c) Interseção e (d) Complemento de A.

Operações generalizadas

As operações nebulosas padrão não são as únicas generalizações possíveis para as operações sobre conjuntos nebulosos. Para cada uma das três operações, existe uma classe de funções que podem ser utilizadas na generalização dessas operações. As funções que podem ser utilizadas na generalização das operações de interseção e união são referidas na literatura como *t-normas* e *t-conormas* (Klement *et al.*, 2000), respectivamente.

A interseção de conjuntos nebulosos pode ser realizada usando operadores que pertencem a uma classe de funções conhecidas como *t-normas* (Klir & Yuan, 1995). As *t-normas* são operações binárias da forma $i : [0,1]^2 \rightarrow [0,1]$ que satisfazem as seguintes propriedades:

- Comutatividade: $i(x,y) = i(y,x)$;
- Associatividade: $i(x,i(y,z)) = i(i(x,y),z)$;
- Monotonicidade: se $x \leq y$ e $z \leq w$, então $i(x,z) \leq i(y,w)$;
- Condição Limite: $i(0,x) = 0$ e $i(1,x) = x$.

Alguns exemplos de *t-normas* são:

- Intersecção Padrão: $a \mathbf{t}_1 b = \min\{a, b\}$
- Produto Algébrico: $a \mathbf{t}_2 b = ab$
- Diferença Limitada: $a \mathbf{t}_3 b = \max\{0, a + b - 1\}$

Logo, interseção nebulosa generalizada ($A \cap B$) dos conjuntos A e B é definida por:

$$(A \cap B)(x) = A(x) \mathbf{t} B(x)$$

onde \mathbf{t} denota uma *t-norma*.

A união de conjuntos nebulosos pode ser realizada usando operadores que pertencem a uma classe de funções conhecidas como *s-normas* ou *t-conormas* (Klir & Yuan, 1995). As *s-normas* são operações binárias da forma $u : [0,1]^2 \rightarrow [0,1]$ que satisfazem as seguintes propriedades:

- Comutatividade: $u(x,y) = u(y,x)$;
- Associatividade: $u(x,u(y,z)) = u(u(x,y),z)$;
- Monotonicidade: se $x \leq y$ e $z \leq w$, então $u(x,z) \leq u(y,w)$;
- Condição Limite: $u(0,x) = x$ e $u(1,x) = 1$.

Alguns exemplos de s-normas são:

- União Padrão: $a \mathbf{s}_1 b = \max\{a, b\}$
- Soma Algébrica: $a \mathbf{s}_2 b = a + b - ab$
- Soma Limitada: $a \mathbf{s}_3 b = \min\{1, a + b\}$

Logo, a união nebulosa generalizada $(A \cup B)$ dos conjuntos A e B é definida por:

$$(A \cup B)(x) = A(x) \mathbf{s} B(x)$$

onde \mathbf{s} denota uma s-norma.

Devido à propriedade de associatividade das t-normas e s-normas, as operações de união e interseção nebulosas podem ser aplicadas a n conjuntos, com $n \geq 2$.

De forma análoga, a operação generalizada de complemento nebuloso pode ser realizada usando-se um operador da classe de operadores definido como $c : [0,1] \rightarrow [0,1]$ que satisfazem as seguintes propriedades:

- Monotonicidade: para todo $x, y \in [0,1]$, se $x \leq y$, então $c(x) \geq c(y)$;
- Condição Limite: $c(0) = 1$ e $c(1) = 0$;
- Involução: $c(c(x)) = x$ para $x \in [0,1]$.

2.3. Relações Nebulosas

As relações convencionais (*crisp*) representam a presença ou ausência de associação, interação ou interconexão entre os elementos de dois ou mais conjuntos.

Uma relação entre os conjuntos *crisp* \mathbf{X} e \mathbf{Y} é um subconjunto do produto cartesiano $\mathbf{X} \times \mathbf{Y}$ e é denotada por $R(x,y)$. Toda relação pode ser definida por uma função característica denotada pelo mesmo símbolo R , como:

$$R: \mathbf{X} \times \mathbf{Y} \rightarrow \{0, 1\}$$

Se $R(x,y) = 1$, dizemos que x e y estão relacionados, ou que o par (x,y) pertence à relação R . Se $R(x,y) = 0$, dizemos que x e y não estão relacionados e o par (x,y) não pertence à relação R .

As relações nebulosas estabelecem associações (interação, interconexão) de diferentes graus entre os elementos pertencentes às relações. A função característica de uma relação *crisp* pode ser generalizada para permitir que pares tenham diferentes graus de pertinência na relação.

Supondo que \mathbf{X} e \mathbf{Y} são dois conjuntos *crisp*, uma relação nebulosa R é um conjunto nebuloso definido no produto cartesiano $\mathbf{X} \times \mathbf{Y}$ que associa a cada elemento do produto cartesiano $\mathbf{X} \times \mathbf{Y}$ um grau de pertinência definido no intervalo unitário, ou seja:

$$R: \mathbf{X} \times \mathbf{Y} \rightarrow [0,1]$$

O valor $R(x,y)$ representa a força (ou o grau) com que os elementos x e y estão relacionados pela R .

Podem-se representar as relações nebulosas da forma:

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

Por exemplo, se tivermos dois universos $\mathbf{X} = \{a, b, c\}$ e $\mathbf{Y} = \{x, y, z\}$, uma relação R sobre $\mathbf{X} \times \mathbf{Y}$ pode ser definida por:

$$R = \{0/(a,x); 0,4/(a,y); 1/(a,z); 0,3/(b,x); 0,9/(b,y); 0/(b,z); 0,5/(c,x); 0,7/(c,y); 0,2/(c,z)\}$$

Tanto as relações *crisp* como as nebulosas não estão restritas a dois conjuntos como descrito acima. No caso geral, uma relação pode estabelecer associação entre três, quatro ou mais conjuntos. Uma relação entre n conjuntos é chamada de relação *n-ária*. Uma relação nebulosa *n-ária* é definida por uma função de pertinência *n-ária* da forma:

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

tal que $R(x_1, x_2, \dots, x_n)$ define o grau com que os elementos da *n-upla* (x_1, x_2, \dots, x_n) estão relacionados pela R .

As relações nebulosas desempenham um papel relevante na definição da semântica das regras nebulosas (detalhes na seção 2.4.3) e sua utilização nas regras de inferência (detalhes na seção 2.4.4).

2.4. Computação com Regras Nebulosas

As regras são utilizadas como um mecanismo formal para representar conhecimento de um domínio. As regras nebulosas, mais especificamente, permitem a representação de conhecimento impreciso, o que muitas vezes torna a modelagem do problema mais fiel e adequada ao mundo real. Todo processo computacional envolvendo regras consiste em aplicar os chamados mecanismos de raciocínio que usam as regras nebulosas juntamente com os fatos, para fazer inferências. Esta seção descreve rapidamente o conceito de variável lingüística, a sintaxe e semântica das regras nebulosas e os fundamentos do processo de inferência e raciocínio baseado em regras.

2.4.1. Variáveis Lingüísticas

Grande parte da experiência e do conhecimento dos seres humanos pode ser mais bem representada na forma lingüística, mais geral e imprecisa. Por outro lado, as representações numéricas são mais precisas e mais facilmente tratáveis por computador. As variáveis lingüísticas permitem que problemas complexos ou mal definidos sejam adequadamente modelados com a representação apropriada das imprecisões naturalmente presentes nesses domínios pelo uso de termos lingüísticos, ao mesmo tempo em que possibilita seu tratamento computacional. Essas variáveis provêm mecanismos de tradução entre descrições lingüísticas imprecisas e descrições numéricas, passíveis de tratamento computacional. Essa tradução se baseia no mapeamento entre termos lingüísticos e conjuntos nebulosos.

Uma variável lingüística pode ser definida, de maneira informal, como uma variável cujos valores são palavras ou sentenças em linguagem natural ao invés de números (Zimmermann, 1991). Elas são definidas sobre um determinado domínio, o qual é granularizado em termos lingüísticos definidos por conjuntos nebulosos. O processo de granularização de um domínio de uma variável em conjuntos nebulosos define a chamada partição nebulosa.

O conjunto de termos lingüísticos que a variável lingüística pode assumir pode conter termos primários e não-primários. Os termos primários são aqueles que representam as categorias mais elementares (por exemplo, *baixa*, *média*, *alta*) e têm uma associação direta com cada conjunto nebuloso da partição definida sobre o domínio da variável lingüística. Os termos não-primários são formados a partir dos termos primários com o uso de modificadores (*muito*, *mais ou menos*, etc.) e conectivos (*e*, *ou não*), como por exemplo, *muito baixa*, *não muito baixa e não muito alta*, etc. Usualmente, as funções de pertinência associadas aos termos primários têm formatos conhecidos (triangular, trapezoidal, Gaussiana – detalhes na seção 2.1.1). As funções de pertinência associadas aos termos não primários são obtidas pela aplicação de operadores sobre as funções dos termos primários. Usualmente, os modificadores são implementados pelas operações de um único argumento e os conectivos, pelas operações de interseção, união e complemento de conjuntos nebulosos (detalhes em Pedrycz & Gomide, 1998).

A figura 2.3 ilustra o exemplo utilizado aqui, em que a variável lingüística temperatura tem como domínio o intervalo $[0, 50]$ e a partição nebulosa foi construída

granularizando esse domínio em três termos lingüísticos *baixa*, *média* e *alta*, da forma trapezoidal.

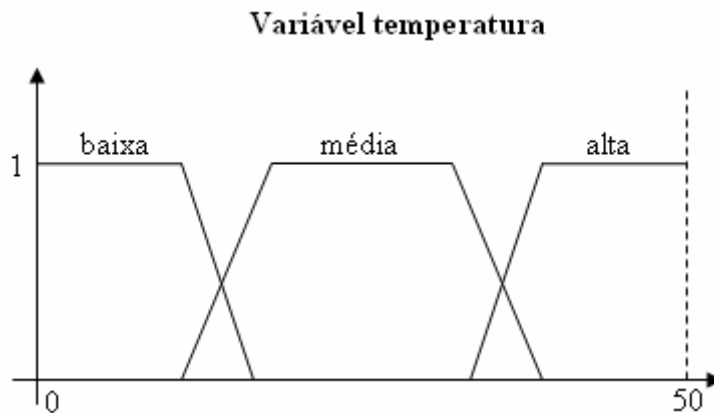


Figura 2.3: Domínio da variável lingüística temperatura e sua granularização.

2.4.2. Sintaxe das Regras Nebulosas

O uso de regras como mecanismo formal para representação do conhecimento remonta às primeiras décadas da criação da área de Inteligência Artificial quando as regras de produção foram propostas como um par *condição-ação* que define uma porção de conhecimento para solução de um problema (Luger, 2002). As regras são uma das técnicas mais antigas para representação de conhecimento e permanecem sendo largamente utilizadas. Sua popularidade vem do fato de possibilitar a expressão clara de diretivas e estratégias, captar o conhecimento experimental de especialistas humanos e possuir um formato lingüístico de fácil compreensão.

No contexto de Sistemas Nebulosos as regras adquirem sintaxe e semântica particulares e são chamadas de regras nebulosas. Sua principal propriedade é a capacidade de representar conhecimento impreciso, pelo uso de termos lingüísticos.

Em geral, as regras nebulosas têm a seguinte forma:

SE <antecedente> **ENTÃO** <conseqüente>

e têm a finalidade de estabelecer relações entre as variáveis que aparecem no antecedente – também chamado de condição ou premissa – e as que aparecem no conseqüente – também chamado de conclusão ou ação. Por exemplo, a regra:

SE a distância é grande **ENTÃO** o tempo de viagem é longo
estabelece uma relação entre as variáveis *distância* e *tempo_de_viagem* usando os termos *grande* e *longo*, que são inerentemente imprecisos.

Com a finalidade de detalhar melhor a sintaxe de uma regra nebulosa, discutimos inicialmente a *proposição atômica nebulosa*. A proposição atômica nebulosa é a parcela de informação básica que pode aparecer em uma regra e seu formato é:

$$X \text{ é } A$$

onde X é uma variável lingüística sobre um domínio \mathbf{X} e A é um termo lingüístico que representa um conjunto nebuloso.

Por exemplo, se X for a variável *distância* e A o termo *grande*, a proposição X é A representa formalmente a parcela de informação do exemplo anterior que diz *a distância é grande*.

Proposições nebulosas podem ser compostas, isto é, construídas a partir de proposições atômicas pelos operadores de conjunção ou disjunção. Por exemplo, as proposições:

$$X_1 \text{ é } A \text{ E } X_2 \text{ é } B$$

e

$$X_1 \text{ é } C \text{ OU } X_2 \text{ é } D$$

podem representar conhecimento do tipo:

A temperatura está alta **E** a umidade está baixa

e

O tempo está nublado **OU** o volume de trabalho é muito grande, respectivamente.

Uma regra nebulosa assume o formato de uma proposição condicional, que no caso mais simples fica:

$$\text{SE } X \text{ é } A \text{ ENTÃO } Y \text{ é } B$$

onde X e Y são variáveis lingüísticas sobre os domínios \mathbf{X} e \mathbf{Y} , respectivamente, e A e B são termos lingüísticos, que rotulam conjuntos nebulosos definidos sobre os universos \mathbf{X} e \mathbf{Y} , respectivamente.

Em uma regra nebulosa, tanto o antecedente quanto o conseqüente podem ser formados por proposições compostas, o que dá origem, por exemplo, a regras da forma:

$$\text{SE } X_1 \text{ é } A_1 \text{ E } X_2 \text{ é } A_2 \text{ E } X_3 \text{ é } A_3 \text{ ENTÃO } Y_1 \text{ é } B_1 \text{ E } Y_2 \text{ é } B_2$$

No caso mais geral, tanto o antecedente como o conseqüente da regra podem ser formados por um número finito de proposições atômicas relacionadas por conjunção ou

disjunção. Assim, assumindo que p e q denotam proposições nebulosas, uma regra nebulosa pode ser da forma **SE p ENTÃO q** sendo que a proposição p pode ter a forma:

$$X_1 \text{ é } A_1 \text{ E } X_2 \text{ é } A_2 \text{ E } \dots \text{ E } X_n \text{ é } A_n, n \geq 1$$

ou

$$X_1 \text{ é } A_1 \text{ OU } X_2 \text{ é } A_2 \text{ OU } \dots \text{ OU } X_n \text{ é } A_n, n \geq 1$$

e a proposição q pode ter a forma:

$$Y_1 \text{ é } B_1 \text{ E } Y_2 \text{ é } B_2 \text{ E } \dots \text{ E } Y_m \text{ é } B_m, m \geq 1$$

ou

$$Y_1 \text{ é } B_1 \text{ OU } Y_2 \text{ é } B_2 \text{ OU } \dots \text{ OU } Y_m \text{ é } B_m, m \geq 1$$

onde A_1, A_2, \dots, A_n e B_1, B_2, \dots, B_m são conjuntos nebulosos nos universos $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n$, $\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_m$, respectivamente, e X_1, X_2, \dots, X_n e Y_1, Y_2, Y_m são variáveis sobre os domínios $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n, \mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_m$, respectivamente.

2.4.3. Semântica das Regras Nebulosas

Passamos agora a discutir a semântica das regras nebulosas. Considerando uma regra nebulosa no seu formato mais simples:

$$\text{SE } X \text{ é } A \text{ ENTÃO } Y \text{ é } B$$

entendemos que esta regra descreve uma relação entre as variáveis X e Y , que pode ser descrita por uma relação nebulosa R cuja função de pertinência $R(x,y)$ descreve o grau com que o par $(x,y) \in \mathbf{X} \times \mathbf{Y}$ é compatível com a relação entre as variáveis X e Y . Dizemos que uma regra nebulosa nessas condições *induz* uma relação nebulosa R .

Sendo A e B conjuntos nebulosos sobre \mathbf{X} e \mathbf{Y} , respectivamente a relação R pode ser determinada por

$$R(x,y) = f(A(x), B(y)), \forall (x,y) \in \mathbf{X} \times \mathbf{Y}$$

onde f é uma função na forma $f : [0,1]^2 \rightarrow [0,1]$.

A escolha da função f é fundamental para o Sistema Nebuloso, pois determina a semântica da regra e diversos aspectos de processamento do sistema.

Em geral as relações induzidas por regras nebulosas derivam de três classes principais de funções: conjunção nebulosa, disjunção nebulosa ou implicação nebulosa (Pedrycz & Gomide, 1998).

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

$$f_t(A(x), B(y)) = A(x) \mathbf{t} B(y), \text{ onde } \mathbf{t} \text{ denota uma t-norma.}$$

As definições mais comuns são as de Mamdani (Mamdani & Assilian, 1975; Mamdani, 1977) e Larsen (Larsen, 1980), que utilizam o mínimo e o produto algébrico.

- Mamdani: $f_m(A(x), B(y)) = A(x) \wedge B(y)$.
- Larsen: $f_p(A(x), B(y)) = A(x) \cdot B(y)$.
- Disjunção Nebulosa é uma função $f_s : [0,1]^2 \rightarrow [0,1]$ definida por:
 $f_s(A(x), B(y)) = A(x) \mathbf{s} B(y)$, onde \mathbf{s} denota uma s-norma.
- Implicação Nebulosa é uma função $f_l : [0,1]^2 \rightarrow [0,1]$ definida por um conjunto de propriedades e classificada em categorias caracterizadas pelo formalismo da lógica clássica e da lógica intuicionista. A definição formal não é apresentada aqui por não ser fundamental para o objetivo do trabalho. Mais detalhes podem ser encontrados em Pedrycz & Gomide (1998). Um dos exemplos mais conhecidos de implicação nebulosa é a implicação de Lukasiewicz (1970):

$$f_l(A(x), B(y)) = \min[1, 1 - A(x) + B(y)]$$

Caso a regra possua múltiplas variáveis no antecedente ou no conseqüente, a relação R é definida de forma semelhante. Para a regra:

$$\mathbf{SE} X_1 \text{ é } A_1 \mathbf{E} X_2 \text{ é } A_2 \mathbf{E} \dots \mathbf{E} X_n \text{ é } A_n \mathbf{ENTÃO} Y_1 \text{ é } B_1 \mathbf{E} Y_2 \text{ é } B_2 \mathbf{E} \dots \mathbf{E} Y_m \text{ é } B_m$$

definimos a relação R assumindo que a proposição composta $X_1 \text{ é } A_1 \mathbf{E} X_2 \text{ é } A_2 \mathbf{E} \dots \mathbf{E} X_n \text{ é } A_n$ induz uma relação **n-ária** definida por:

$$P_a(x_1, x_2, \dots, x_n) = A_1(x_1) \mathbf{t} A_2(x_2) \mathbf{t} \dots \mathbf{t} A_n(x_n)$$

e a proposição composta $Y_1 \text{ é } B_1 \mathbf{E} Y_2 \text{ é } B_2 \mathbf{E} \dots \mathbf{E} Y_m \text{ é } B_m$ induz uma relação **m-ária** definida por:

$$P_c(y_1, y_2, \dots, y_m) = B_1(y_1) \mathbf{t} B_2(y_2) \mathbf{t} \dots \mathbf{t} B_m(y_m)$$

A regra induz então uma relação de ordem $n+m$ definida por:

$$R(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m) = f(P_a(x_1, x_2, \dots, x_n), P_c(y_1, y_2, \dots, y_m))$$

Caso o antecedente ou conseqüente da regra seja formado pelo conectivo **OU** em vez de **E**, substituímos a t-norma por uma s-norma no cálculo das relações induzidas P_a ou P_c .

2.4.4. Inferência com Regras Nebulosas

A computação com regras nebulosas envolve a aplicação de um método de raciocínio, o qual está baseado em regras de inferência. A *regra da inferência composicional* é a regra de inferência básica para os processos de raciocínio nos sistemas nebulosos. A regra da inferência composicional, também chamada de regra da composição, será apresentada para o caso simples, isto é, quando a regra tem apenas uma variável no antecedente e uma no conseqüente.

O esquema de inferência de interesse nesse contexto é aplicado a partir de uma regra, que estabelece um conhecimento genérico sobre algum domínio, e de um fato, que estipula uma informação específica sobre a situação da qual se deseja inferir alguma conclusão.

Seja A um conjunto nebuloso sobre \mathbf{X} , e B um conjunto nebuloso sobre \mathbf{Y} . Conforme o que foi descrito anteriormente sobre a semântica das regras, a regra

$$\mathbf{SE } X \text{ é } A \mathbf{ ENTÃO } Y \text{ é } B$$

induz uma relação nebulosa $R(x,y)$ sobre $\mathbf{X} \times \mathbf{Y}$. Para usar essa regra em uma inferência, é necessário conhecer algum fato que estabeleça um valor específico para a variável do antecedente X .

Dado o fato $X \text{ é } A'$, pode-se inferir a conclusão $Y \text{ é } B'$, sendo que B' é definido por (Pedrycz & Gomide, 1998):

$$B'(y) = \sup_{x \in \mathbf{X}} [A'(x) \mathbf{t} R(x,y)] \text{ para todo } y \in \mathbf{Y}$$

onde \sup denota o operador de supremo.

A aplicação da regra da inferência composicional pode ser facilmente estendida para regras com múltiplas variáveis, e também para situações em que o conhecimento do problema é descrito por um conjunto de regras. O processo geral de computação usando regras nebulosas é baseado na regra de inferência composicional.

Alternativamente a este, pode ser aplicado o chamado método de inferência escalonada, que envolve um número pequeno de cálculos numéricos com relação ao método geral. Esses dois processos gerais de computação com regras estão descritos em Pedrycz & Gomide (1998). O processo geral de inferência e a inferência escalonada não serão apresentados aqui, uma vez que não são utilizados no sistema estudado. A discussão apresentada focaliza nos métodos de raciocínio específicos para problemas de classificação, que é o foco principal deste trabalho.

2.5. Sistemas Nebulosos Baseados em Regras

Um Sistema Nebuloso é qualquer sistema que incorpora algum mecanismo derivado da lógica nebulosa e que pelo menos uma das variáveis do problema pode assumir valores lingüísticos definidos por conjuntos nebulosos (Zadeh, 1973; Klir & Yuan, 1995).

O tipo particular de Sistemas Nebulosos de interesse para este trabalho são os Sistemas Nebulosos Baseados em Regras (SNBR) ou simplesmente Sistemas de Regras Nebulosas. Os SNBR têm sido usados com sucesso em diversas áreas de aplicação, principalmente em problemas de controle, classificação e modelagem (ver, por exemplo: Pedrycz, 1996; Pedrycz & Gomide 1998; Dumitrescu *et al.*, 2000). Um SNBR é composto por dois componentes principais: a Base de Conhecimento (BC) e o Mecanismo de Inferência (MI).

A Base de Conhecimento é composta pela Base de Dados Nebulosa (BDN), a qual contém as definições dos conjuntos nebulosos relacionados aos termos lingüísticos usados na regras nebulosas e pela Base de Regras Nebulosas (BRN), que armazena o conjunto de regras nebulosas que modelam um determinado problema.

O Mecanismo de Inferência é responsável pelo processamento das regras, o qual é realizado por um método de raciocínio. Este consiste da aplicação de um procedimento de inferência para derivar conclusões a partir das regras e de fatos conhecidos (seção 2.4.4).

Com a definição dos diversos parâmetros envolvidos em um SNBR, obtem-se o que é chamado de modelo de sistema nebuloso, sendo que alguns desses modelos recebem destaque na literatura, devido a seu sucesso em aplicações práticas. Nesse sentido, podemos citar dois modelos principais de sistemas nebulosos: os de **Mamdani** (Mamdani & Assilian, 1975; Mamdani, 1977) que possuem na parte conseqüente das regras nebulosas uma proposição nebulosa e os de **Takagi-Sugeno** (Takagi & Sugeno, 1983; 1985) que possuem na parte conseqüente uma função aplicada sobre os valores de entrada.

Na próxima subsecção descrevemos com detalhes os sistemas nebulosos de classificação, que são o tipo de sistema de interesse para este trabalho.

2.5.1. Sistemas Nebulosos de Classificação

Classificação é uma importante tarefa encontrada nas áreas de reconhecimento de padrões, tomada de decisão, mineração de dados e modelagem (Berg *et al.*, 2002). Dado um conjunto de objetos $E = \{e_1, e_2, \dots, e_l\}$, também chamados de *padrões*, o objetivo da classificação é atribuir uma classe C_j de um conjunto conhecido de classes $C = \{C_1, C_2, \dots, C_m\}$ a um objeto $e_p = \{a_{p1}, a_{p2}, \dots, a_{pn}\}$, o qual é descrito por n atributos.

Muitos métodos têm sido utilizados para a tarefa de classificação de padrões, tais como árvores de decisão (Quinlan, 1986), métodos estatísticos (Duda & Hart, 1973), redes neurais (Bishop, 1995) e sistemas nebulosos (Kecman, 2001).

Um sistema nebuloso de classificação é um SNBR que tem a finalidade de definir a classe a que um determinado padrão pertence.

Uma típica regra nebulosa de classificação segue o formato geral das regras nebulosas descrito na seção 2.4.2 e pode ser expressa por:

$$R_k : \mathbf{SE} X_1 \text{ é } A_{i1} \mathbf{E} \dots \mathbf{E} X_n \text{ é } A_{in} \mathbf{ENTÃO} \text{ Classe} = C_j$$

sendo R_k o identificador da regra, X_1, \dots, X_n os atributos do padrão considerado no problema (representados aqui por variáveis lingüísticas), A_{i1}, \dots, A_{in} os valores lingüísticos usados para representar os valores de tais atributos e C_j a classe, nebulosa ou não, a que pertence o padrão.

O **Mecanismo de Inferência** aplica o conjunto de regras nebulosas no padrão a ser classificado, determinando a classe a que ele pertence. A maioria dos sistemas nebulosos de classificação utiliza o *Método de Raciocínio Nebuloso Clássico* (Chie *et al.*, 1996; Abe & Thawonmas, 1997; Gonzalez & Perez, 1999), que classifica um padrão usando a regra que possuir maior grau de compatibilidade com esse padrão.

Considere o padrão a ser classificado $e_p = \{a_{p1}, a_{p2}, \dots, a_{pn}\}$ e um conjunto de regras $R = \{R_1, R_2, \dots, R_s\}$. O raciocínio nebuloso clássico se dá pelas seguintes etapas:

- Calcular o grau de compatibilidade, $Compat(R_k, e_p)$, entre o padrão e_p e cada regra R_k , $k = 1, \dots, s$, usando uma t-norma sobre o grau de pertinência dos valores dos atributos do padrão, a_{pj} , aos correspondentes conjuntos nebulosos que aparecem no antecedente da regra, A_{ij} , $j = 1, \dots, n$.

$$Compat(R_k, e_p) = \mathbf{t}(A_1(a_{p1}), A_2(a_{p2}), \dots, A_n(a_{pn}))$$

- Encontrar a regra que possui maior grau de compatibilidade com o padrão.

$$\max\{ Compat(R_k, e_p) \}, k = 1, \dots, s$$

- Ao padrão e_p será atribuída a classe C_j , tal que C_j é a classe da regra R_k que possui o maior grau de compatibilidade com o padrão.

Esta forma de raciocínio usa somente uma regra para classificar o padrão e desperdiça as informações providas por outras regras, embora elas também possuam um grau de compatibilidade com o padrão.

Sendo assim, um outro método de raciocínio que combina as informações providas por todas as regras para classificar o padrão pode ser utilizado. Ele recebe o nome de *Método de Raciocínio Nebuloso Geral* e se dá pelos seguintes passos (Ishibuchi *et al.*, 1999a):

- Calcular o grau de compatibilidade, $Compat(R_k, e_p)$, entre o padrão e_p e cada regra R_k , $k = 1, \dots, s$, usando uma t-norma sobre o grau de pertinência dos valores dos atributos do padrão, a_{pj} , aos correspondentes conjuntos nebulosos que aparecem no antecedente da regra, A_{ij} , $j = 1, \dots, n$.

$$Compat(R_k, e_p) = \mathbf{t}(A_1(a_{p1}), A_2(a_{p2}), \dots, A_n(a_{pn}))$$

- Calcular para cada classe C , $C = 1, \dots, m$, o grau de compatibilidade $Classe_C$ do padrão dado com a classe C , agregando os graus de compatibilidade do padrão com todas as regras que o classificam nessa classe C , como segue:

$$Classe_C = \sum \{ Compat(R_k, e_p) \mid C \text{ é a classe da regra } R_k \}$$

- O padrão e_p será classificado na classe C , sendo C a classe que possuir maior valor na equação do passo anterior.

Este trabalho apresenta uma abordagem baseada em co-evolução para construir automaticamente os sistemas nebulosos de classificação.

Capítulo 3

Computação Evolutiva

Em 1859, Darwin (1859), naturalista inglês, apresentou o conceito de *Seleção Natural*, princípio segundo o qual os indivíduos mais adaptados ao meio apresentam maior possibilidade de sobreviver e gerar descendentes. Este princípio é resultado da observação de que as mais diferentes formas de vida são suscetíveis à adaptação, que ocorre por meio de lentas transformações genéticas, conforme os indivíduos evoluem. O processo de evolução ocorre através de ciclos fixos nas gerações, ou seja, cada indivíduo nasce, cresce, normalmente gera um ou mais descendentes e morre.

A computação evolutiva (CE) é uma área da ciência da computação que abrange modelos computacionais inspirados na *Teoria da Evolução das Espécies*, essencialmente no conceito de *Seleção Natural*, para a solução de problemas, nas mais diversas áreas do conhecimento. Nos anos de 1950 e 1960, muitos biólogos começaram a desenvolver simulações computacionais de sistemas genéticos. A partir desse período diversas abordagens surgiram, sendo que algumas ganharam maior credibilidade no meio científico. Mais recentemente foi cunhado o termo Computação Evolutiva para agregar os diversos trabalhos voltados para a simulação de aspectos da evolução.

A evolução, de uma forma geral, é um processo de otimização. A Computação Evolutiva tem sido aplicada com essa finalidade a diversas áreas, como, por exemplo, planejamento, projeto, simulação e identificação, controle e classificação (Beasley, 2000).

Conforme apontado em Bittencourt (2006), esse paradigma difere de outros modelos computacionais convencionais por não exigir, para resolver o problema, o

conhecimento prévio de uma maneira de encontrar a solução. São sistemas que tentam resolver problemas acumulando conhecimento sobre o problema e utilizando essas informações para gerar soluções aceitáveis, com base em conceitos de adaptação.

Sistemas baseados em CE, também frequentemente referidos como Algoritmos Evolutivos (AE), se caracterizam por envolver reprodução, variação aleatória, competição e seleção de indivíduos em uma população. Tais algoritmos partem do pressuposto que, em uma dada população, indivíduos com boas características genéticas têm maiores chances de sobrevivência e de produzirem indivíduos cada vez mais aptos. Como resultado, os indivíduos menos aptos tenderão a desaparecer.

Os AE possuem alguns componentes comuns, que são descritos aqui, com base no que foi apresentado em Bäck (2000), com o objetivo de formular a estrutura básica desses algoritmos:

- Uma população de indivíduos que representam possíveis soluções para o problema e que são normalmente representadas como pontos no espaço de busca; o conjunto inicial de soluções pode ser aleatório ou pode ser obtido a partir de técnicas convencionais para resolver instâncias simples do problema que está sendo tratado;
- Uma função de avaliação projetada para avaliar a aptidão de cada um dos indivíduos com relação ao meio. De acordo com esse valor de aptidão, um processo de seleção é aplicado de forma a permitir que indivíduos mais bem avaliados se reproduzam com maior frequência;
- Um conjunto de operadores que serão aplicados à população de uma dada geração para obter os indivíduos da próxima geração. Esses operadores são baseados nos fenômenos que ocorrem na evolução natural como seleção, mutação e recombinação. Operadores genéticos que garantem a diversidade dos indivíduos dessa população ao longo da evolução.

As diferentes classes de sistemas da CE utilizam esses componentes de diferentes maneiras e combinações.

Para definir a função de avaliação é necessário encontrar uma maneira de codificar as soluções, cujo resultado corresponde aos cromossomos na evolução natural e é chamado de genótipo. A partir desses cromossomos, a função de avaliação deve ser capaz de determinar a qualidade de uma solução.

Na subseção seguinte são descritas resumidamente as diversas técnicas que compõem a CE e em seguida são apresentados com mais detalhes, os Algoritmos Genéticos, que é a classe de AE utilizada neste trabalho.

3.1. Técnicas de Computação Evolutiva

A área de CE é mais conhecida pelas suas subáreas que, de acordo com o agrupamento proposto por Bäck (2000) são: Algoritmos Genéticos, Estratégias Evolutivas e Programação Evolutiva. Em Fogel (1995) e Bäck (1996) pode-se encontrar uma revisão detalhada de similaridades e diferenças dessas técnicas.

Os AG foram propostos inicialmente por John Holland (1975), que começou a desenvolver pesquisas sobre simulações computacionais de sistemas genéticos. Holland foi gradualmente refinando suas idéias e em 1975 publicou o seu livro *Adaptation in Natural and Artificial Systems*, hoje considerado referência básica dos Algoritmos Genéticos. Esses algoritmos passaram a ser aplicados nos mais diversos problemas de otimização e Aprendizado de Máquina (Goldberg, 1989). Os AG utilizam a recombinação como o operador mais importante e aplicam mutação com uma probabilidade bem baixa.

As Estratégias Evolutivas (EE) foram propostas na Alemanha (Rechemberg, 1965) com a finalidade de resolver problemas hidrodinâmicos e de controle. É uma técnica utilizada principalmente por pesquisadores da Engenharia Civil (Carvalho *et al.*, 2003). Utilizam mutações com distribuições normais para modificar vetores de números reais e enfatizam mutação e recombinação como operadores essenciais para busca no espaço de soluções e no espaço de parâmetros ao mesmo tempo. O operador de seleção é determinístico e as populações de pais e descendentes geralmente têm número diferente (Bäck, 2000).

A Programação Evolutiva (PE), desenvolvida originalmente por Lawrence Fogel *et al.* (1966) e refinada posteriormente por David Fogel (1995), enfatiza a mutação e não aplica a recombinação de indivíduos. Assim como nas estratégias de evolução utiliza mutação com distribuição normal e estende o processo evolutivo aos parâmetros da estratégia. O operador de seleção é probabilístico e atualmente a maioria das aplicações relatadas usa espaços de busca de vetores de números reais, mas o algoritmo foi inicialmente proposto para evolução de máquinas de estados finitos (Bäck, 2000).

Outro ramo da CE, considerado descendente dos Algoritmos Genéticos, é o da Programação Genética (PG) (Koza, 1992). Na PG, os indivíduos de uma população são programas de computador em vez de cadeias de valores binários ou reais. Esses programas são representados por árvores que podem apresentar tamanhos e formas diferentes. A avaliação de cada indivíduo da população ocorre por meio da execução do programa representado pelo indivíduo. Geralmente, PG não utiliza operadores de mutação.

Apesar de terem origens bastante diversas, todas essas abordagens têm em comum o modelo conceitual inicial, que é a evolução natural.

Na seção seguinte são apresentados os Algoritmos Genéticos, que pertencem à classe de AEs de interesse para este trabalho.

3.2. Algoritmos Genéticos

Como já dito na seção anterior, os Algoritmos Genéticos, propostos por Holland (1975), são algoritmos de otimização global baseados nos mecanismos de seleção natural e da genética. Apesar de serem algoritmos de busca aleatória, os AG não empregam busca aleatória desprovida de direção, mas exploram informação histórica para especular novos pontos de busca dos quais é esperado melhor desempenho (Goldberg, 1989). Isso é feito por meio de um processo iterativo, tal que em cada iteração uma nova geração de indivíduos é produzida.

A principal distinção entre os AGs e as outras técnicas de CE é a ênfase na recombinação genética, ou cruzamento.

O termo Algoritmos Genéticos refere-se, na verdade, a uma classe de algoritmos que se baseiam em passos genéricos com uma estrutura bastante simples, e que assumem versões variadas de acordo com a aplicação ou com particularidades da implementação.

O funcionamento básico de um AG pode ser explicado como segue. Inicialmente é gerada uma população de indivíduos, que usualmente é aleatória, representando possíveis soluções para o problema. Durante o processo evolutivo, esta população é avaliada e a cada indivíduo é atribuído um valor de aptidão, que deve refletir sua habilidade de adaptação a um determinado ambiente (qualidade da solução). Uma porcentagem dos mais adaptados é mantida, enquanto os outros são descartados. Os

membros mantidos pela seleção podem sofrer modificações em suas características fundamentais através das operações de mutação e recombinação genética, também chamada de cruzamento (crossover), gerando descendentes para a próxima geração. Este processo, chamado de reprodução, é repetido até que uma solução satisfatória seja encontrada (Carvalho *et al.*, 2003).

O ponto de partida para a utilização de AG como ferramenta para solução de problemas é a representação destes problemas de maneira adequada. Isso exige a formulação de uma codificação para cada indivíduo da população, que usualmente utilizam vetores de tamanho finito em um alfabeto finito.

Neste ponto pode ser útil detalhar a terminologia utilizada. As estruturas que representam os indivíduos são chamadas de cromossomos e, usualmente, são cadeias de símbolos. Esses são os genótipos que são manipulados pelo AG. O processo de cálculo do valor de aptidão decodifica essas estruturas em um fenótipo, que são as representações do problema, e atribuem um valor a ele. Na versão clássica de AG, também chamado de AG canônico, cada indivíduo da população é representado por um conjunto de elementos, ou seja, um vetor, no qual cada elemento (gene), em uma posição específica (*locus*), pode assumir dois valores (alelos). Nas formulações mais recentes, outras formas de codificação passaram a ser utilizadas, com base em um conjunto maior de valores possíveis para os genes, sendo a mais difundida a codificação de números reais.

A escolha da função de aptidão utilizada no AG é de fundamental importância para a obtenção de um bom resultado do processo. A função é definida levando em conta o problema específico, e deve ser capaz de avaliar cada indivíduo (candidato à solução) de forma a atribuir valores numéricos que forneçam uma estimativa da qualidade desse indivíduo, ou seja, o quanto esse indivíduo se aproxima da solução considerada ideal. O valor obtido pela função de aptidão é o principal critério usado pelo operador de seleção, para determinar quais indivíduos vão ser utilizados pelos demais operadores na construção da geração seguinte.

Uma questão importante para a aplicação dos AG é o papel dos operadores de mutação e recombinação. Eles são utilizados para assegurar que a nova geração seja realmente nova, mas possua, de alguma forma, características de seus pais, ou seja, a população se diversifica e mantém características de adaptação adquiridas pelas gerações anteriores.

Basicamente, os algoritmos genéticos possuem a estrutura apresentada no algoritmo descrito na figura 3.1 a seguir (Michalewicz, 1996), em que $P(t)$ denota a população na geração t .

Algoritmo

início

$t \leftarrow 0$;

iniciar $P(t)$;

avaliar $P(t)$;

enquanto condição de parada não for satisfeita **faça**

$t \leftarrow t + 1$;

selecionar $P(t)$ de $P(t-1)$;

aplicar cruzamento em $P(t)$;

aplicar mutação em $P(t)$;

avaliar $P(t)$;

fim enquanto

fim

Figura 3.1: Algoritmo básico de um AG.

O desempenho de um AG é fortemente influenciado pela definição dos parâmetros a serem utilizados, sendo os mais importantes: tamanho da população, número de iterações, com o respectivo critério de parada, e taxas de cruzamento e mutação. Normalmente esses parâmetros são ajustados de forma empírica, por meio de execuções dos algoritmos implementados e devem levar em conta tanto as necessidades do problema quanto os recursos disponíveis.

Ao contrário dos demais esquemas de Computação Evolutiva apresentados inicialmente, os AG aplicam-se a um escopo mais amplo que a simples otimização: são apresentados como um modelo de Aprendizado de Máquina (AM). Na linguagem de Aprendizado de Máquina, os AG constituem uma abordagem para aprendizado em que os indivíduos representam hipóteses e o processo de busca procura pela melhor hipótese, definida como aquela que otimiza uma medida numérica previamente definida do problema em questão. Por exemplo, se o problema de aprendizado for a aproximação de uma função a partir de dados de entrada e saída da função, a aptidão pode ser

definida como a acuidade da hipótese sobre os dados de treinamento. AG têm sido aplicados com sucesso a diversas tarefas de aprendizado, entre elas a geração de uma coleção de regras para problemas de controle, ou a otimização da topologia e aprendizado de parâmetros de redes neurais artificiais (Mitchell, 1996).

O trabalho apresentado nesta dissertação utiliza os AG como mecanismo de aprendizado para construção de Sistemas Nebulosos, como será descrito nos capítulos seguintes.

A seguir são apresentados de forma mais detalhadas os componentes básicos de um AG: formas de codificação de indivíduos e operadores de seleção, mutação e cruzamento.

3.2.1. Codificação

A codificação é a representação de um indivíduo da população. Essa representação é uma estrutura de dados que identifica o cromossomo. A codificação pode ser:

- Codificação Binária: onde cada gene do cromossomo pode assumir os valores 0 (zero) ou 1 (um);
- Codificação Inteira: onde cada gene do cromossomo pode assumir valores inteiros;
- Codificação Real: onde cada gene do cromossomo pode assumir valores reais;
- Codificação Mista: onde cada gene do cromossomo pode assumir uma codificação, inteira ou real;
- Codificação em um Alfabeto: onde cada gene do cromossomo pode assumir valores pertencentes a um alfabeto de símbolos.

O AG clássico foi inspiração para todas as abordagens que se seguiram, embora a codificação binária seja uma restrição para alguns tipos de problemas (Michalewicz, 1996). As novas propostas adotam populações de indivíduos representados por estruturas de dados mais genéricas, nas quais cada componente pode assumir qualquer valor.

3.2.2. Métodos de Seleção

O princípio básico do funcionamento dos AGs é que um critério de seleção vai fazer com que, depois de muitas gerações, o conjunto inicial de indivíduos gere indivíduos mais aptos. Para que o método de seleção aplicado privilegie os indivíduos mais aptos, deve levar em conta o valor calculado pela função de aptidão. Assim, cada indivíduo tem uma probabilidade de ser selecionado proporcional à sua aptidão. Vários métodos de seleção têm sido propostos. A preocupação fundamental na definição de um operador de seleção é favorecer indivíduos com maiores notas de aptidão, mas não exclusivamente, a fim de manter a diversidade da população. A seguir serão apresentados alguns métodos de seleção encontrados na literatura.

Seleção Proporcional à Aptidão (Roleta)

Método apresentado por Holland (1975), onde a probabilidade de um indivíduo ser aceito é diretamente proporcional à sua aptidão, ou seja, indivíduos mais aptos têm maior chance de serem escolhidos para criar os descendentes.

Este esquema pode ser visualizado como uma roleta, onde cada indivíduo tem uma parte dessa roleta proporcional à sua aptidão. Ao se determinar uma parte dessa roleta, indivíduos mais aptos têm mais chances.

A figura 3.2(a) exemplifica esse método.

Esse método é ruim quando os indivíduos têm suas medidas de desempenho próximas, o que faria como se fosse a seleção aleatória. Além disso, se a população é pequena, os indivíduos mais adaptados serão sempre escolhidos levando a uma convergência prematura, não explorando o potencial dos AG.

Amostragem Universal Estocástica

Este operador de seleção foi apresentado por Baker (1987) e é semelhante ao método da roleta. Mas ao invés de escolher um indivíduo por vez que a roleta é virada, escolhe-se mais do que um (dois indivíduos, ou quatro, por exemplo). Isso pode diminuir a convergência prematura em populações pequenas. Porém, em populações

com indivíduos com adaptações parecidas, esse método não apresenta vantagens com relação ao anterior, a seleção é como se fosse aleatória.

A figura 3.2(b) nos mostra esse método, onde um mesmo indivíduo é escolhido mais de uma vez.

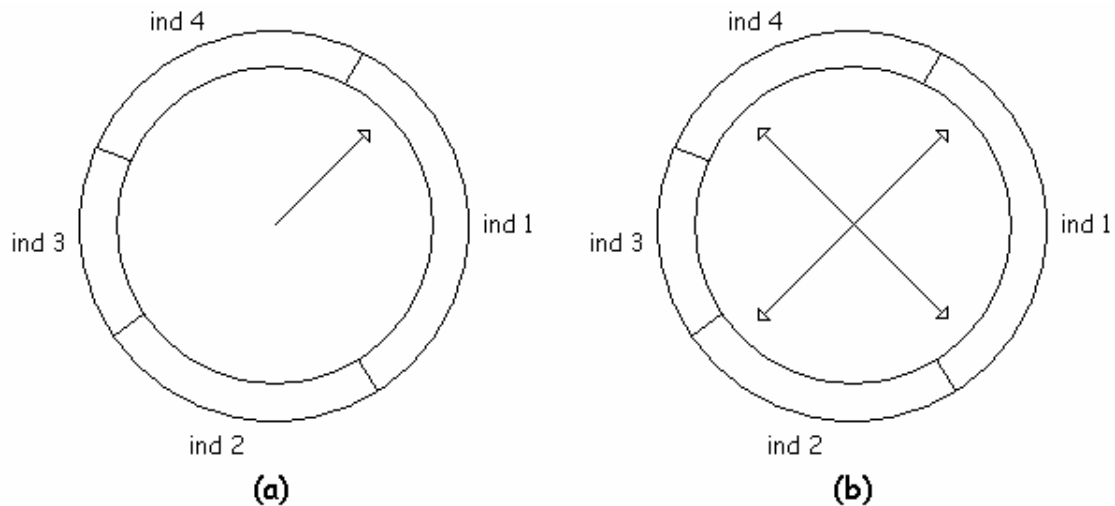


Figura 3.2: (a) Seleção proporcional ao desempenho e (b) Amostragem universal estocástica.

Seleção Baseada na Classificação (Rank)

Uma opção para se evitar super-indivíduos e a ocorrência de convergência prematura é reduzir as diferenças entre os indivíduos supondo uma classificação para cada indivíduo ao invés de pegar diretamente sua medida de desempenho.

Esse método de seleção ordena os indivíduos usando a sua aptidão, e a probabilidade de seleção é atribuída de acordo com a posição dessa ordenação (Gonzalez & Perez, 1999), ou seja, o primeiro indivíduo tem uma probabilidade maior que o segundo, que tem uma maior probabilidade que o terceiro e assim por diante.

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

O torneio (DeJong, 1975; Goldberg, 1989) consiste basicamente em escolher dois indivíduos aleatoriamente da população e um valor x no intervalo de $[0,1]$. Se o

valor x for maior que 0.5 (limiar) o indivíduo mais adaptado é escolhido para a próxima geração senão o indivíduo menos adaptado é escolhido para a nova geração.

Existem variações desse método, por exemplo:

- Um torneio não “justo”, onde o limiar é menor que 0.5 para favorecer os indivíduos mais adaptados;
- Fazer uma escolha determinística, onde sempre o melhor indivíduo é escolhido;
- O torneio não é feito apenas com dois indivíduos, mas sim com uma porcentagem da população.

Elitismo

O termo elitismo, introduzido por De Jong (1975), está associado com uma técnica que pode ser adicionada a qualquer método de seleção, que tem como finalidade descartar a possibilidade dos melhores indivíduos de uma população não estarem na geração seguinte. Consiste em manter o melhor, ou alguns dos melhores indivíduos da população na geração seguinte. Isso garante que esses indivíduos não sejam perdidos ou alterados pelos operadores genéticos.

Em grande parte dos casos, estratégias elitistas associadas aos métodos de seleção melhoram o desempenho dos 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 e/ou recombinação. 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 do mais diverso é baseado em algum critério de distância previamente definido.

3.2.3. Operadores Genéticos

O princípio básico dos operadores genéticos é transformar a população através de sucessivas gerações, estendendo a busca até chegar a um resultado satisfatório. A transformação, chamada de reprodução na terminologia de AGs, é realizada por um conjunto de operadores que mutam e recombina membros selecionados da população corrente. Os operadores genéticos são responsáveis por garantir a diversidade da população ao mesmo tempo que mantêm as características de adaptação adquiridas pelas gerações anteriores.

O mecanismo para produzir variações mais conhecido é a mutação, em que um componente do cromossomo (alelo) é escolhido aleatoriamente e alterado. Em outras palavras, novas soluções são criadas por meio de pequenas mudanças aleatórias feitas na representação de soluções já presentes na população (Bäck, 2000). O cruzamento é o operador responsável pela recombinação de características dos pais durante a reprodução, permitindo que as próximas gerações herdem essas características. Ele é considerado o operador predominante e deve ser aplicado com uma taxa maior que a de mutação. A idéia básica do operador de cruzamento é que, dados dois indivíduos com alta aptidão, criam-se novos indivíduos que combinam as características desses, de forma aleatória. A seguir são descritos operadores de cruzamento e mutação específicos.

Recombinação Genética

A recombinação genética troca parte da informação genética entre indivíduos, produzindo novas soluções potenciais com algumas características dos reprodutores. Primeiramente são selecionados os reprodutores que irão gerar os descendentes, e o cruzamento dos reprodutores será feito com certa taxa, a taxa de cruzamento. Essa taxa representa a probabilidade de cruzamento dos reprodutores. Ou seja, nas novas gerações é possível que os reprodutores selecionados não cruzem, permanecendo eles mesmos na próxima geração, do contrário, seus descendentes é que farão parte da nova geração.

O cruzamento dos reprodutores para gerar os descendentes, pode ser feito de várias maneiras. As formas mais comuns são:

- O cruzamento de um ponto, onde um *locus* é escolhido aleatoriamente, e o material genético é trocado após esse ponto;
- O cruzamento de múltiplos pontos, onde vários *loci* são escolhidos, e a troca do material genético é intercalada entre esses pontos;
- O cruzamento uniforme, onde alguns *loci* são escolhidos aleatoriamente e são trocados os seus alelos.

Existe também o cruzamento aritmético apresentado por Michalewicz (1996). Esse operador é aplicado em cromossomos de codificação real. No cruzamento de dois cromossomos, quatro descendentes são gerados. Se os reprodutores são representados por $C_v = \{c_{v1}, \dots, c_{vn}\}$ e $C_w = \{c_{w1}, \dots, c_{wn}\}$, e os descendentes são C_1, C_2, C_3 e C_4 , podemos calcular os genes dos descendentes da seguinte maneira:

- $c_{1i} = a * c_{wi} + (1 - a) * c_{vi}$
 - $c_{2i} = a * c_{vi} + (1 - a) * c_{wi}$
 - $c_{3i} = \min\{c_{vi}, c_{wi}\}$
 - $c_{4i} = \max\{c_{vi}, c_{wi}\}$
- para $a \in [0,1]$ e $i = 1, \dots, n$.

O parâmetro a pode ser constante ou variar de geração em geração. Desses quatro indivíduos gerados, somente os dois melhores farão parte da população.

A figura 3.3 mostra exemplos das diferentes formas de cruzamento e a figura 3.4 mostra o cruzamento aritmético com o parâmetro a igual a 0.35.

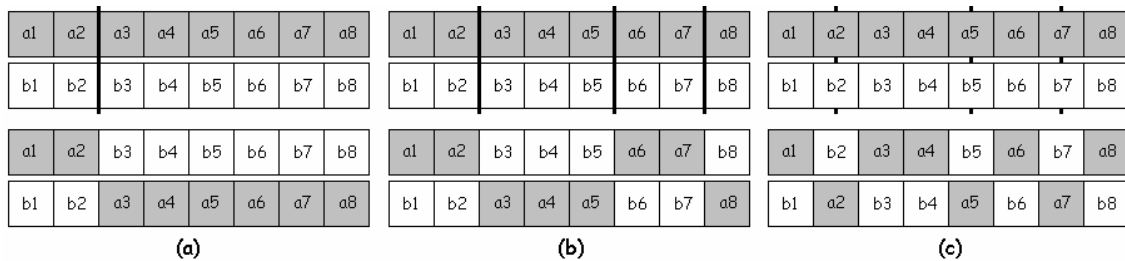


Figura 3.3: Cruzamento: (a) Simple, (b) Múltiplos pontos e (c) Uniforme.

Reprodutores					Descendentes						
C_v :	2	7	8	6	4	C_1 :	2,35	7,70	6,95	4,25	2,60
C_w :	3	9	5	1	0	C_2 :	2,65	8,30	6,05	2,75	1,40
						C_3 :	2	7	5	1	0
						C_4 :	3	9	8	6	4

Figura 3.4: Cruzamento aritmético ($\alpha=0,35$).

Mutação

A mutação altera um ou mais genes de um cromossomo. A taxa de mutação define com que probabilidade será modificado o alelo. É uma operação que tem por finalidade introduzir e manter a diversidade genética na população. Uma taxa muito alta pode tornar o algoritmo desprovido de uma direção em sua pesquisa pelo espaço de busca. Já uma taxa de mutação muito baixa pode deixar o processo de busca lento. Normalmente, as taxas são 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. Dessa forma a mutação evita que a população pare de evoluir.

O processo de mutação é implementado com base em um operador unário que transforma o valor de um gene. Existem três tipos mais comuns de mutação: simples, inteira e real (Michalewicz, 1996). Na mutação simples, usada na codificação binária, troca-se o valor do bit, ou seja, se o bit vale 0 (zero), troca-o por 1 (um), ou se o bit vale 1 (um), troca-o por 0 (zero). Na mutação inteira, o valor do gene é alterado por um número inteiro pertencente a um alfabeto de valores inteiros. Na mutação real, o operador de mutação pode realizar uma busca ampla sobre um domínio real ao longo de todo o processo evolutivo para a escolha do novo valor para o gene, como é o caso da mutação uniforme, ou realizar uma busca mais abrangente no início da evolução e mais localizada nas gerações finais, como é o caso da mutação não-uniforme.

A figura 3.5 ilustra exemplos de cromossomos com diferentes tipos de codificação e de tipos de operadores de mutação.

1	0	1	0	1	0	0	1	7	0	6	2	3	8	1	4	0,2	0,0	0,6	0,7	0,3	0,8	0,1	0,4
1	1	1	0	1	0	0	1	7	0	6	7	3	8	1	4	0,2	0,0	0,6	0,7	0,9	0,8	0,1	0,4
(a)								(b)								(c)							

Figura 3.5: Mutação: (a) Simples, (b) Inteira e (c) Real uniforme.

Na figura 3.5(a), a mutação simples ocorre na segunda posição e troca o alelo 0 para 1. A figura 3.5(b) traz um exemplo de mutação inteira na quarta posição que altera o alelo 2 para 7. Na figura 3.5(c), a mutação real uniforme ocorre na quinta posição e altera o alelo para um novo valor dentro de um domínio $D_k = [0,1]$.

3.3. Co-Evolução

A área de computação evolutiva foi enriquecida nos últimos anos com uma nova ramificação, que consiste de abordagens que utilizam os conceitos de co-evolução, e deu origem à chamada computação co-evolutiva. Co-evolução pode ser vista como uma extensão do modelo evolutivo tradicional, onde duas ou mais espécies (populações) interagem constantemente entre si por meio de um processo de avaliação acoplado dentro do meio ambiente (ecossistema). Assim, problemas complexos podem ser decompostos em subproblemas (subcomponentes – espécies), onde cada subproblema terá uma contribuição na solução do problema como um todo. Essa forma de evolução favorece a descoberta de soluções complexas para problemas que possam ser decompostos naturalmente (Paredis, 1995).

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 (Delgado, 2002).

A abordagem co-evolutiva possibilita a aplicação de algoritmos inspirados nos fenômenos da seleção natural a um escopo maior de problemas, que são aqueles com

características que dificultam ou inviabilizam seu tratamento pelos algoritmos evolutivos de uma única população.

De acordo com Pena-Reyes & Sipper (2001), as abordagens co-evolutivas são mais adequadas do que os algoritmos evolutivos de uma única população quando o problema a ser resolvido apresenta uma ou mais das seguintes características: a solução sendo procurada é complexa; o problema ou a sua solução podem claramente ser decompostos; o cromossomo codifica tipos diferentes de valores; forte interdependência entre os componentes da solução; a ordenação dos componentes afeta drasticamente o valor de aptidão.

Muito embora os algoritmos co-evolutivos sejam de certa forma mais poderosos que as versões tradicionais, algumas questões adicionais relacionadas com a decomposição do problema devem ser tratadas com cuidado quando se deseja utilizar essa abordagem estendida. Para se aplicar esses algoritmos com sucesso em problemas com complexidade cada vez maior, torna-se necessário introduzir noções explícitas de modularidade nas soluções para que elas disponham de oportunidades razoáveis de evoluir na forma de sub-componentes co-adaptados (Cruz, 2003).

Potter & De Jong (2000) apresentam uma discussão clara a esse respeito e colocam que as questões a serem resolvidas são a decomposição do problema em si, a dependência entre os componentes no processo de evolução, a contribuição de cada espécie na solução e a manutenção da diversidade.

Ainda de acordo com Potter & De Jong (2000), a decomposição do problema consiste em determinar um número apropriado de sub-componentes e o papel de cada um. A questão da evolução dos sub-componentes interdependentes diz respeito à modelagem apropriada da interdependência entre componentes para garantir que aconteça uma co-adaptação. A terceira questão, relativa à contribuição de cada componente na solução final consiste na avaliação correta, muitas vezes não óbvia, da influência de um componente em particular, que pode ser positiva ou negativa na obtenção de uma solução global. Finalmente, a questão de manutenção da diversidade, que nas versões tradicionais de CE é resolvida por mecanismos muito simples, requer atenção especial, uma vez que, mesmo que um indivíduo de uma população seja o mais forte para contribuir na solução final, os demais não poderiam ser descartados enquanto não se encontrar a melhor solução com a presença de todos os sub-componentes.

Genericamente falando, pode-se dizer que espécies que co-evoluem podem competir ou cooperar, o que caracteriza as duas grandes classes de abordagens propostas na literatura: as abordagens competitivas e as abordagens cooperativas.

Em um algoritmo co-evolutivo competitivo a aptidão de um indivíduo é baseada na competição direta com indivíduos de outras espécies, que evoluem separadamente nas suas respectivas populações. O aumento do valor de aptidão de uma espécie implica na diminuição da aptidão das outras espécies. Essa forma de evolução, na situação ideal, deve incrementar as capacidades de cada espécie até atingir um ponto máximo (Pena-Reyes & Sipper, 2001).

Como exemplo de abordagens competitivas, podemos citar predador/presa e hospedeiro/parasita. No caso de predador/presa, a co-evolução visa predadores mais perigosos e presas com estratégias de defesa mais eficazes (Geffeney *et al.*, 2005). Dessa forma, a co-evolução de ambas as espécies visa selecionar os predadores com boas técnicas de predação e as presas com grande capacidade e evitar a predação. Já no exemplo hospedeiro/parasita, a co-evolução visa defesas mais efetivas dos hospedeiros para que os parasitas possam evoluir com maior mordacidade.

Alguns dos trabalhos mais difundidos de abordagem evolutiva competitiva foram propostos em (Hillis, 1991; Rosin & Belew, 1997), em que foram usadas versões hospedeiro/parasita por envolver duas populações em que indivíduos de cada população se alternam nos papéis de testar ou serem testados. Os autores usam o termo hospedeiro para se referir ao indivíduo cuja aptidão está sendo calculada e parasita para se referir aos indivíduos que estão testando o hospedeiro.

Algoritmos co-evolutivos cooperativos, também chamados simbióticos, envolvem duas ou mais espécies que evoluem independentemente, mas colaboram e interagem na busca pela solução de problemas, em um mesmo meio ambiente, muitas vezes referido como eco-sistema na terminologia de Sistemas Co-evolutivos. Cada espécie tem seu próprio esquema de codificação e evolução, assim não há troca de material genético. A aptidão de um indivíduo depende da sua capacidade de cooperar com as demais espécies na construção da solução global.

3.3.1. Abordagens Co-Evolutivas

Nesta subseção serão descritos resumidamente os principais trabalhos sobre co-evolução que fundamentam a abordagem adotada no trabalho descrito aqui. As propostas de abordagens co-evolutivas especificamente para a geração de Sistemas Nebulosos enquadram-se no tema de Sistemas Nebulosos Genéticos e serão descritas no Capítulo 4.

O trabalho de Giordana e outros (1994) e seus desdobramentos (Giordana & Néri, 1996), projetados especificamente para aprender regras de classificação, são considerados precursores da aplicação do mecanismo de co-adaptação de sub-componentes, embora não utilizem o conceito de múltiplas espécies. A decomposição do problema é tratada por um operador de seleção que faz um agrupamento de indivíduos. Uma solução completa é formada pela seleção da melhor regra em cada agrupamento.

Um dos primeiros trabalhos que utilizaram modelos com múltiplas espécies, ou seja, populações geneticamente isoladas, foi apresentado por Hillis (1990), onde é proposto um modelo co-evolutivo competitivo do tipo hospedeiro/parasita para gerar redes de ordenação. Nesse trabalho, uma espécie – a hospedeira - representa as redes de ordenação e a outra espécie – a parasita – representa os casos de testes na forma de seqüências de números a serem ordenados.

A versão hospedeiro/parasita foi também adotada como modelo básico em Rosin & Belew (1997), em que novos métodos para co-evolução competitiva foram explorados no contexto de aprendizado de estratégias para jogos como Nim e Jogo da Velha tri-dimensional. Três técnicas de co-evolução competitiva foram exploradas, as quais os autores deram os nomes de compartilhamento de adaptação competitiva, compartilhamento de amostras e hall da fama.

Olsson (2001) apresentou novos resultados relativos à abordagem competitiva, motivados pela necessidade identificada pelo autor de uma maior formalização de algoritmos do tipo hospedeiro/parasita, praticamente inexistente nos trabalhos anteriores. Para isso, ele explora o papel do problema de assimetria no processo de evolução, sendo que se entende por assimetria a grande diferença entre o tipo de indivíduos e tamanho do espaço de busca das duas espécies envolvidas. A principal contribuição da proposta é um ambiente de avaliação de algoritmos do tipo hospedeiro/parasita.

Um número significativo de trabalhos tem sido dedicado à construção de redes neurais artificiais por meio da co-evolução. Entre esses, podemos citar a proposta de Moriarty & Miikkulainen (1997, 1998), que usa uma abordagem colaborativa com duas espécies. Cada indivíduo de uma das espécies corresponde a um único neurônio de camada intermediária da rede, e suas conexões com as camadas de entrada e saída. Os indivíduos da outra espécie codificam conjuntos de neurônios de camadas intermediárias, ou seja, conjuntos de indivíduos da primeira população, para formar uma rede.

Uma abordagem diferente é proposta por Garis (1990), com o objetivo de gerar redes neurais para controlar criaturas simuladas. O processo usado por Garis consiste em definir uma decomposição do problema de controle manualmente, e aplicar AG para evoluir pequenas redes neurais especializadas em cada tarefa.

Uma das abordagens cooperativas mais significativas, no sentido em que inspiraram propostas que surgiram posteriormente, foi a de Paredis (1995). Paredis aplicou co-evolução cooperativa a problemas que envolviam a busca simultânea por valores de uma solução e sua ordem adequada. Nessa proposta, uma população de soluções evoluía de forma cooperativa juntamente com uma população de permutações aplicadas aos cromossomos que representavam as soluções.

No trabalho apresentado por Potter & De Jong (2000), é proposto um modelo de co-evolução cooperativa que, ao contrário das abordagens cooperativas conhecidas até então, enfatiza a decomposição emergente do problema. Esse conceito relaciona-se a visão dos autores de que a preocupação principal ao procurar extensões dos paradigmas evolutivos tradicionais é com a divisão do problema em sub-componentes que “emergem” durante o próprio processo evolutivo, em vez de serem projetados manualmente. O principal destaque do trabalho de Potter & De Jong está no fato de que eles descrevem uma arquitetura generalizada para co-evolução adaptada de componentes que interagem, chamada co-evolução cooperativa, que modela um ecossistema consistindo de duas ou mais espécies. Não há restrições quanto ao número de espécies e qualquer técnica de computação evolutiva pode ser aplicada, desde que adequada para o problema sendo tratado.

As abordagens descritas até este ponto são consideradas, em geral, referências básicas nos demais trabalhos encontrados na literatura e, de uma forma ou de outra, influenciaram a proposta investigada nesta dissertação.

Diversos outros trabalhos relativos a algoritmos co-evolutivos continuaram a surgir e alguns deles são relatados resumidamente na seqüência, com o objetivo de destacar a grande atividade existente na área, embora não tenham tido influência direta no trabalho descrito aqui.

Bucci & Pollack (2002) apresentaram um arcabouço teórico para o estudo da co-evolução, baseado na matemática dos conjuntos ordenados. Esse arcabouço é usado para descrever soluções para problemas de otimização co-evolutiva.

A abordagem proposta por Ficici *et al.* (2003) apresenta um mecanismo de memória co-evolutiva, com base em princípios da teoria de jogos, chamado de *memória Nash*. A motivação dos autores foi prevenir o problema de *esquecimento* observado em algoritmos co-evolutivos, em que comportamentos adquiridos previamente são perdidos por não serem usados.

Um outro trabalho apresentado em (De Jong *et al.*, 2003) aborda o problema de definição da função de avaliação. Considerando que nos AE a construção da função de avaliação é dinâmica, por ser baseada na interação entre indivíduos, os autores questionam se a extensão de tal avaliação é adequada. Nesse trabalho é definida a noção de uma função de avaliação ideal e é mostrado que a co-evolução pode, em princípio atingir a avaliação ideal.

O trabalho apresentado em (Bucci et al, 2004) investiga como a noção de estrutura do problema pode se tornar precisa e propõe uma definição formal de estrutura do problema. Baseado nessa definição é introduzido um procedimento automático para extração da estrutura do problema, que por sua vez dá um significado preciso à noção de objetivos subjacentes a um problema, conceito este que tem sido usado para explicar como devem ser avaliados os indivíduos no contexto co-evolutivo.

Os métodos de seleção dos algoritmos co-evolutivos são avaliados em (Ficici *et al.*, 2005) com o uso da Teoria de Jogos Co-evolutiva (EGT). Os resultados obtidos enfatizam o grau com que o comportamento do modelo é sensível a detalhes de implementação dos métodos de seleção.

Em (Bucci & Pollack, 2005) os autores propõem uma modificação aos Algoritmos Co-evolutivos Cooperativos para favorecer a busca por um ótimo global. As modificações propostas baseiam-se num mecanismo da co-evolução de Pareto (De Jong, 2004) que compara indivíduos usando a dominância de Pareto, conceito fundamental da área de otimização.

A questão da avaliação dos indivíduos é também abordada por De Jong & Bucci (2006), que propõem um algoritmo chamado *Dimension Extracting Coevolutionary Algorithm (DECA)*. O problema atacado nesse trabalho vem do fato de que os algoritmos co-evolutivos usam medidas de agregação entre os valores de aptidão de indivíduos de uma população, como por exemplo, a média, para encontrar o valor de aptidão de indivíduos de outra população e assim aplicar os métodos de seleção, sendo que essa forma de avaliação é instável. As abordagens propostas para resolver o problema de instabilidade baseiam-se no armazenamento de um grande volume informações o que pode se tornar inviável em certas aplicações. O algoritmo proposto utiliza um processo de avaliação que reduz a informação necessária ao cálculo dos valores de aptidão.

No próximo capítulo serão apresentados os Sistemas Nebulosos Genéticos, com as principais abordagens propostas, com destaque para aquelas utilizam conceitos de co-evolução na construção de Sistemas Nebulosos.

Capítulo 4

Sistemas Nebulosos Genéticos

O indiscutível sucesso dos Sistemas Nebulosos Baseados em Regras em diversas áreas de aplicação tem sido enfatizado frequentemente nos últimos anos. Uma das principais motivações para o uso desse tipo de sistemas está na habilidade de representar conhecimento inerentemente impreciso, presente em quase todos os domínios do conhecimento do mundo real.

Além da reconhecida utilidade e importância dos SNBR, os trabalhos reportados na literatura ressaltam também a importância do processo de construção desses sistemas. A construção de um SNBR, também denominada Modelagem Nebulosa (Yager & Filev, 1994) exige atenção especial, pois dela depende o bom desempenho do sistema. Essa etapa abrange a construção das regras nebulosas e dos conjuntos nebulosos que compõem a partição dos domínios envolvidos no problema.

A construção de sistemas capazes de adquirir conhecimento de forma automática, ou seja, por meio de um programa de computador, é objeto de estudo da área de Aprendizado de Máquina (Resende, 2003). Dizemos que um sistema que adquire conhecimento automaticamente é um sistema que dispõe de capacidade de aprendizado.

A geração automática de SN, seja exclusivamente de conjuntos de dados ou combinando conjuntos de dados com conhecimento de especialistas tem sido o foco de interesse de muitos pesquisadores e tem se tornado uma abordagem promissora na construção de sistemas robustos. As razões para esse interesse crescente na geração automática de tais sistemas são, principalmente, o imenso volume de dados disponível

em praticamente todas as áreas do conhecimento, a confiança adicional associada aos sistemas obtidos a partir de dados reais e a possibilidade de aplicação dos métodos desenvolvidos a problemas de dimensões maiores, que não poderiam ser tratados por métodos manuais.

Os SN não dispõem de capacidades inerentes de aprendizado, logo, métodos de construção automática desses sistemas baseiam-se na combinação da metodologia dos SN com outras metodologias que possuam recursos de aprendizado, o que resulta em sistemas híbridos. As abordagens mais bem sucedidas nesse sentido foram as tentativas de hibridação feitas no contexto da computação flexível (*Soft Computing*, ver Pedrycz, W., 1998) utilizando os mecanismos de aprendizado das redes neurais artificiais (Jang *et al.*, 1997; Nauck & Kruse, 1997; Castellano & Fanelli, 2000) e dos algoritmos genéticos (AG) (Pedrycz, 1997; Sanchez *et al.*, 1997; Cordón *et al.*, 2004b).

Durante os últimos 15 anos, os AGs têm sido usados na tarefa de definição da Base de Conhecimento dos SNBR com diferentes objetivos, como por exemplo: definição das regras nebulosas, redução do número de regras nebulosas pré-estabelecidas, aperfeiçoamento dos conjuntos nebulosos por otimização de parâmetros, eliminação de redundâncias, entre outras.

De uma maneira geral, cunhou-se o termo Sistemas Nebulosos Genéticos (SNG) para nomear um Sistema Nebuloso combinado com um processo de aprendizado baseado em Algoritmos Genéticos. Este trabalho focaliza um tipo particular de SNG, que são aqueles sistemas que utilizam AGs para gerar SNBR, chamados de Sistemas Nebulosos Genéticos Baseados em Regras (SNGBR).

A classe dos SNG inclui também as Redes Neurais Nebulosas Genéticas e os Algoritmos de Agrupamento Nebulosos Genéticos, que não serão abordados aqui por estarem fora do escopo do trabalho. Uma coleção significativa e abrangente de modalidades e de novas tendências de SNGBR pode ser encontrada em (Cordón *et al.*, 2004a).

Com o uso de AG como método de aprendizado, o projeto automático de SN é modelado como um processo de otimização ou busca em um espaço de soluções em potencial, que representam partes ou o todo de uma Base de Conhecimento. Nesse caso dizemos que o Sistema Nebuloso foi obtido por um processo de aprendizado evolutivo.

A Modelagem Nebulosa estabelece dois requisitos para o modelo obtido: interpretabilidade (ou clareza) – capacidade de expressar o comportamento do sistema real de uma maneira compreensível – e acuidade (ou precisão) – capacidade de

representar o sistema real com fidelidade. Entretanto, obter altos graus de interpretabilidade e acuidade é um objetivo contraditório e, na prática, uma das duas propriedades prevalece sobre a outra. No decorrer dos últimos anos, reconheceu-se não ser apropriado negligenciar uma ou outra propriedade, o que originou uma tendência que vem crescendo em importância na comunidade científica, de procurar um balanceamento entre interpretabilidade e acuidade.

Quando se usa AG para gerar um SN, os critérios adotados para caracterizar a qualidade do resultado procurado devem determinar a definição da função de aptidão sendo aplicada. Com esse objetivo, é necessário compreender com clareza quais parâmetros do sistema indicam uma boa acuidade e uma boa interpretabilidade.

A determinação da acuidade ou precisão é mais facilmente entendida já que acuidade significa a propriedade do sistema obter respostas que efetivamente correspondam ao esperado. Nesse caso, a função de aptidão do AG deve basear-se em medidas de erro ou distância a uma saída esperada.

Quando se trata de interpretabilidade, sua caracterização depende de vários aspectos. Não é possível identificar uma definição precisa de interpretabilidade, mas diversos trabalhos apresentam discussões sobre essa questão (Jin, 2000; Setnes & Roubos, 2000; Casillas *et al.*, 2005).

Para os objetivos do trabalho apresentado aqui vamos considerar a interpretabilidade em função dos três fatores apresentados por Yaochu Jin (Jin, 2000). O primeiro fator relaciona-se à distribuição das funções de pertinência. Para que um sistema seja compreendido, a partição nebulosa deve ser completa e distinguível, isto é, todo o domínio de uma variável deve ser coberto por algum conjunto nebuloso e esses conjuntos não devem ter muita semelhança nem serem em número muito alto. Assim, é possível dar um significado semântico para cada conjunto, por meio de um termo lingüístico. O segundo fator é que a base de regras deve ser consistente, uma vez que regras contraditórias dificultam a compreensão do seu significado. O terceiro fator está relacionado ao número de variáveis no antecedente de cada regra e com o número total de regras na base. O pressuposto básico nesse caso é que quanto menor for o número de variáveis e o número de regras, mais fácil será compreender o significado das regras.

Caso o objetivo do projeto automático do SNBR seja obter sistemas com boa interpretabilidade, a função de avaliação deve incluir medidas que representem esses fatores relacionados a ela.

Este trabalho considera a questão de interpretabilidade relacionada aos três fatores descritos acima, por meio da definição da função de aptidão correspondente à população ou por meio de restrições na aplicação de operadores. O número de variáveis no antecedente das regras e o número de regras são levados em consideração nas funções de aptidão dos respectivos AGs. A distribuição dos conjuntos nebulosos é considerada pela definição de limites superiores e inferiores para os intervalos de variação dos parâmetros das funções e a questão da consistência é considerada na utilização de operadores aplicados na população de bases de regras. Mais detalhes são apresentados no próximo capítulo.

4.1. Abordagens de Sistemas Nebulosos Genéticos

O tópico de Sistemas Nebulosos Genéticos Baseados em Regras, apesar de razoavelmente recente, já deu origem a um número expressivo de trabalhos e, como área de pesquisa, continua em crescente atividade nos dias de hoje. Apesar da dificuldade em classificar as abordagens recentes que surgem a cada ano trazendo enfoques inovadores, já é possível identificar algumas consideradas clássicas, bem como novas tendências.

A organização das abordagens de SNGBR existentes em categorias está fundamentada na definição de dois aspectos básicos: qual parte da base de conhecimento é o objeto de otimização pelo AG e qual classe de problemas está sendo abordada – problemas de ajuste ou adaptação de elementos da base de conhecimento previamente definidos ou problemas de construção ou projeto de partes da base de conhecimento sem definição prévia.

A seguir são apresentadas as principais abordagens encontradas na literatura para projeto ou otimização de Sistemas Nebulosos Baseados em Regras, agrupadas em uma tentativa de classificação extraída e adaptada de Cordón *et al.* (2004b) e Cordón *et al.* (2001c).

No contexto dos SNGBR os métodos que combinam as abordagens nebulosa e genética para geração de Bases de Conhecimento podem ser divididos em dois grupos principais: métodos que fazem o ajuste ou sintonia de componentes da Base de Conhecimento (Adaptação Genética) e métodos que constroem componentes da Base de Conhecimento (Construção Genética).

4.1.1. Adaptação Genética

Estão incluídos no grupo de Adaptação Genética os métodos que partem da Base de Regras Nebulosa (BRN) ou Base de Dados Nebulosa (BDN) existentes e usam AG para aperfeiçoar o desempenho do sistema fazendo ajuste ou adaptação de uma ou mais partes da Base de Conhecimento (BC). Esses métodos podem ser subdivididos em grupos de acordo com o enfoque adotado:

Adaptação Genética dos Conjuntos Nebulosos

Neste grupo encontram-se os métodos que focalizam a sintonia de conjuntos nebulosos (Karr, 1991a ; Herrera *et al.*, 1995; Bonissone *et al.*, 1996; Cordón & Herrera 1997; Gurocak 1999; Casillas *et al.*, 2005). Nesses métodos a BRN e a BDN são previamente definidas, manualmente ou por um processo automático qualquer. A BRN permanece fixa e a BDN é ajustada pelo AG, por meio da alteração dos parâmetros das funções de pertinência de todas as partições nebulosas envolvidas no problema. Em Casillas *et al.* (2005), particularmente, é proposta a alteração dos conjuntos nebulosos com a finalidade de sintonia combinando a alteração dos parâmetros das funções de pertinência com a aplicação de modificadores lingüísticos aos conjuntos originais. As partições nebulosas são, portanto representadas no cromossomo de cada indivíduo por meio de seus parâmetros, que variam conforme o tipo de função de pertinência sendo usado (seção 2.1.1).

Otimização Genética do número de regras

Esta categoria inclui os métodos que otimizam o número de regras de uma BRN, eliminando redundâncias (Ishibuchi *et al.*, 1997; Cordón & Herrera, 1997). Nessa classe de métodos, um conjunto prévio de regras é obtido por um mecanismo automático – genético ou não – e posteriormente o AG é aplicado para selecionar, a partir do conjunto inicial de regras, um subconjunto de regras que mantenha o mesmo desempenho, eliminando as regras desnecessárias.

4.1.2. Construção Genética

O grupo chamado aqui de construção genética engloba os métodos que usam AG para construir (ou projetar) efetivamente um ou mais componentes da BC. A definição prévia de um dos componentes da BC pode ser necessária se o objetivo do processo genético for a geração do outro componente da BC. Esse grupo de métodos foi o que gerou o maior número de pesquisas e pode ainda ser subdividido em três subgrupos, dependendo da parte da Base de Conhecimento que vai ser projetada usando AG: a Base de Regras Nebulosas, a Base de Dados Nebulosa ou ambas. Os subgrupos são descritos a seguir.

Construção Genética da Base de Regras

Essa abordagem focaliza a construção das regras nebulosas, considerando um conjunto de funções de pertinência já definido e fixo. Usualmente a definição da Base de Dados é feita escolhendo-se um número de valores lingüísticos para as variáveis lingüísticas e distribuindo uniformemente os valores, que são rótulos dos conjuntos nebulosos, no domínio de cada variável (Wang & Mendel, 1992; Hwang & Thompson, 1994; Ishibuchi *et al.*, 1994; Hoffmann & Pfister, 1997; Gonzalez & Perez, 1999; Casillas *et al.*, 2000; Cordón & Herrera, 2000).

Nesse grupo enquadram-se algumas das abordagens mais utilizadas e conhecidas, identificadas como:

- A abordagem de Pittsburgh (Hoffmann & Pfister, 1997; Spiegel & Sudkamp, 2003), cuja principal característica é que cada cromossomo codifica um conjunto completo de regras, ou seja, toda a base de regras nebulosas. Assim, cada indivíduo é um candidato à solução. A principal desvantagem desse tipo de abordagem é que o crescimento do tamanho do cromossomo para problemas com número elevado de variáveis pode inviabilizar sua utilização. Por outro lado, essa forma de representação pode tirar proveito da natural competição que ocorre entre indivíduos de uma população do AG, por representar a base de regras completa.
- A abordagem de Michigan (Bonarini, 1996; Ishibuchi *et al.*, 1999b), na qual cada cromossomo codifica apenas uma única regra e a base de

regras é formada por toda a população. Em oposição à abordagem anterior, neste caso não ocorre o problema de aumento excessivo do tamanho do cromossomo, porém, a questão da competição entre as regras (indivíduos) deve ser tratada com cuidado, uma vez que as regras de uma base de regras devem cooperar entre si e não competir.

- A abordagem iterativa (Cordón & Herrera, 1997; Gonzalez & Perez, 1999; Jesus *et al.*, 2004; Hoffmann, 2004), na qual cada cromossomo codifica apenas uma única regra, como na abordagem de Michigan. Neste caso, porém, o processo genético é intercalado com um processo iterativo em que, a cada nova população gerada pelo AG, a melhor regra da população é selecionada para fazer parte da base de regras. Esse processo é repetido até que a base de regras inclua regras suficientes para obter o desempenho desejado.

Construção Genética da Base de Dados

Nessa abordagem, proposta por Cordón *et al.* (2001a), o AG é usado para gerar a Base de Dados Nebulosa sem nenhum tipo de definição prévia, possivelmente modificando a cada passo o número de conjuntos nebulosos em cada partição bem como os parâmetros que controlam a definição das funções. O conjunto de regras também não é definido previamente, mas é reconstruído por meio de um método de geração automática de regras para cada indivíduo de cada população, em todas as gerações. Cada indivíduo representa uma possível partição para o problema e, a cada geração, um novo conjunto de regras é definido para cada um desses indivíduos pelo processo de geração de regras adotado, para que a função de adaptação (*fitness*) do AG possa ser calculada. Essa abordagem não é uma abordagem clássica como as anteriores e difere da adaptação genética dos conjuntos nebulosos na qual o que é feito é um ajuste de conjuntos considerando um conjunto fixo de regras.

Construção Genética da Base de Conhecimento

Os métodos nesse subgrupo usam AG para definir as regras e os conjuntos nebulosos simultaneamente (Lee & Takagi, 1993a; Parodi & Bonelli, 1993; Carse *et al.*,

1996; Filipic & Juricic, 1996; Ishibuchi & Murata, 1996; Magdalena & Monasterio, 1997; Velasco, 1998; Cordón & Herrera, 2001; Rivas *et al.*, 2003; Angelov, 2003; Kim *et al.*, 2006). A complexidade do processo de busca nesse caso é alta, uma vez que o espaço no qual a busca vai acontecer é uma composição do espaço das possíveis regras com o espaço das possíveis funções de pertinência. Tanto as regras quanto os conjuntos nebulosos são representados na codificação.

Além das abordagens clássicas descritas nos grupos apresentados, um grande número de outras propostas surgiu nos últimos anos e continuam a surgir até os dias de hoje, trazendo abordagens que representam refinamentos ou combinações das categorias já detalhadas e outras que consistem em propostas mais inovadoras e sofisticadas. Devido à grande variedade de propostas e o curto período de tempo transcorrido desde seu surgimento, não é possível traçar uma distinção clara entre características de cada método, não sendo, portanto, apropriado agrupá-las em categorias como foi feito com as abordagens mais conhecidas. Na seqüência são destacadas algumas das abordagens mais recentes.

Alguns trabalhos enfocam a geração de toda Base de Conhecimento – Base de Bados e Base de Regras –, não de forma simultânea como já descrito anteriormente e sim de forma seqüencial, aplicando inicialmente um processo genético para geração das regras nebulosas e em seguida um outro processo genético para ajustar os conjuntos nebulosos. Exemplos desse tipo de abordagem podem ser encontrados em Chiou & Lan (2005) e Cordón & Herrera (1997).

Em Ishibuchi & Yamamoto (2004) é proposta a utilização de AG para selecionar as regras que vão compor a BRN para problemas de classificação a partir de um conjunto de regras candidatas. O método consiste de duas fases: primeiro as regras candidatas são geradas a partir de conjuntos de dados numéricos pelo uso das medidas de avaliação de regras da mineração de dados (confiança e suporte); depois um número pequeno de regras é selecionado a partir desse conjunto inicial, usando um algoritmo genético multi-objetivos. Esse método tem a finalidade de atacar o problema da dimensionalidade do espaço de busca para o processo genético da segunda fase, reduzindo o número de regras em potencial, por algum critério de pré-seleção das regras que serão consideradas pelo AG.

Diversas propostas combinam processos de aprendizado baseados em AG com outros métodos de aprendizado. Em Juang (2005) é apresentada uma combinação de

agrupamento com AG baseado em aprendizado por reforço. Inicialmente, a parte das condições das regras é gerada por uma técnica *on-line* de agrupamento chamado de alinhamento baseado em agrupamento. Em seguida, os conseqüentes são gerados por um processo de aprendizado genético por reforço, baseado no método de valor-Q (Mitchell, 1996), que funciona como o valor de aptidão do AG. Aprendizado por reforço é um método de aprendizado de máquina em que o sistema deve aprender algum comportamento por meio de interações com um ambiente dinâmico, baseadas em tentativa e erro. Mais detalhes podem ser encontrados em Berenji & Khedkar (1992).

O trabalho apresentado em Lee & Shin (2003) propõe um algoritmo adaptativo de mínimos quadrados pela combinação de mínimos quadrados com AG, para fazer o aprendizado híbrido da estrutura e parâmetros de sistemas nebulosos sem intervenção humana. O sistema nebuloso nesse caso é representado por uma rede neural-nebulosa, que combina noções da inferência nebulosa e de redes neurais (Jang & Misutani, 1997).

As abordagens descritas em Jesus *et al.* (2004) e Hoffmann (2004) combinam a abordagem já descrita anteriormente chamada de abordagem iterativa com algoritmo de *boosting* para geração de classificadores nebulosos. A base de regras nebulosa é gerada de maneira incremental e o mecanismo de *boosting* reduz o peso das instâncias de treinamento que são classificadas corretamente pela nova regra gerada. Assim, o ciclo seguinte de geração de regra prioriza a geração de regras compatíveis com os exemplos ainda não classificados pelas regras existentes.

O trabalho descrito por Chen & Huang (2003) usa AG para gerar regras nebulosas com pesos a partir de sistemas de base de dados relacional (Codd, 1970), especificamente com o objetivo de estimar valores nulos. Os atributos que aparecem nos antecedentes das regras têm pesos diferentes, os quais são codificados nos cromossomos e ajustados pelo AG.

Em Wang *et al.* (2005) é proposto um novo esquema baseado em AG hierárquico multi-objetivo para extrair regras interpretáveis a partir de dados. O objetivo da proposta é construir modelos Takagi-Sugeno (citado na seção 2.5) considerando a precisão e interpretabilidade. Inicialmente, um método de agrupamento nebuloso é usado para gerar os antecedentes das regras do modelo inicial. Em seguida, o método de mínimos quadrados recursivo é aplicado para determinar os conseqüentes das regras. Aí então esse modelo inicial é usado pelo AG hierárquico multi-objetivo para gerar um sistema nebuloso otimizado, quando é usada a formulação proposta pelos autores de

cromossomo hierárquico, o que permite a otimização simultânea dos antecedentes das regras e do número de regras.

Um aspecto importante a ser destacado nas propostas de SNG que têm surgido na literatura é a preocupação cada vez mais crescente com a questão do balanceamento entre precisão e interpretabilidade. Esse enfoque pode ser verificado, por exemplo, nos trabalhos de Wang *et al.* (2005), Kim *et al.* (2006) e Teng & Wang (2004). Wang *et al.* (2005) e seus co-autores propõem um esquema baseado em AG hierárquico multi-objetivo para extrair regras interpretáveis, usando uma formulação de cromossomo que estrutura de forma hierárquica genes de controle e genes de parâmetros. A função de avaliação engloba conceitos de interpretabilidade e precisão. A proposta apresentada em Kim *et al.* (2006) consiste de um AG com uma nova forma de codificação e uma nova função de avaliação para geração de modelos Takagi-Sugeno. O esquema de codificação proposto consiste de três cromossomos que permitem a otimização simultânea dos parâmetros das funções de pertinência dos antecedentes das regras e da estrutura das regras. A função de avaliação proposta direciona o processo de busca evolutiva de tal forma que podem ser encontrados modelos nebulosos mais precisos e compactos mantendo antecedentes interpretáveis. Em Teng & Wang (2004) o balanceamento entre número de regras (entendido como uma medida de clareza do sistema) e a precisão de aproximação é tratado também pela proposta de uma função de avaliação que considera, além da precisão do conjunto de regras, o número total de funções de pertinência utilizadas.

Outra abordagem, distinta das anteriores, foi proposta Chang & Lilly (2004), na qual é apresentado um método automático de projeto de sistemas nebulosos para classificação por meio da otimização genética. Diferentes sistemas nebulosos são definidos para o mesmo conjunto de dados usando um algoritmo proposto pelos próprios autores chamado VISIT, pela atribuição aleatória de parâmetros iniciais do sistema. O AG encontra o SN ótimo pela otimização dos referidos parâmetros. Esse método permite a geração de um sistema de classificação nebuloso compacto sem nenhum conhecimento *a priori* nem suposições sobre a distribuição inicial dos dados.

Na próxima seção destacamos as abordagens baseadas no conceito de co-evolução, por consistirem no fundamento do trabalho investigado nesta dissertação.

4.1.3. Abordagens Baseadas em Co-Evolução

Os algoritmos co-evolutivos são técnicas evolutivas avançadas para resolver problemas complexos que podem ser decompostos em subproblemas. Eles envolvem duas ou mais espécies (populações) que interagem constantemente por meio de um processo de avaliação acoplado (Cordón *et al.*, 2004b). Tais algoritmos têm sido investigados nos anos mais recentes no contexto de geração automática de Sistemas Nebulosos, como uma das alternativas mais promissoras para resolver problemas relacionados à dimensão do cromossomo e dimensão do espaço de busca encontrados nas abordagens genéticas mais tradicionais.

Os algoritmos co-evolutivos cooperativos propostos por Potter & De Jong (2000) tornaram-se fonte de inspiração para diversos outros trabalhos desenvolvidos posteriormente, incluindo-se os que visam a construção de SN. Em Pena-Reyes & Sipper (2001) foi proposta uma abordagem cooperativa co-evolutiva para gerar sistemas nebulosos onde duas espécies co-evolutivas são definidas: uma consistindo da BDN (funções de pertinência) e outra consistindo da BRN. Cada uma dessas espécies forma uma população que evolui por meio de um AG simples e independente. O cálculo da função de aptidão de um determinado indivíduo de uma das espécies é feito por meio da cooperação de um ou mais representantes da outra espécie. Assim, o Sistema Nebuloso final é construído de forma cooperativa.

A abordagem apresentada por Delgado em (Delgado *et al.*, 2001; 2002; 2004) propõe um método co-evolutivo hierárquico colaborativo para projetar Sistemas Baseados em Regras Nebulosas onde indivíduos de diferentes espécies representam parâmetros do modelo Takagi–Sugeno. Essa abordagem assume que espécies representam soluções parciais de problemas de modelagem nebulosa organizadas em quatro níveis hierárquicos. Os indivíduos de cada nível hierárquico representam funções de pertinência, regras individuais, bases de regras e sistemas nebulosos, respectivamente. A cooperação entre indivíduos de diferentes espécies se dá pela aptidão compartilhada, uma vez que a aptidão de um indivíduo é calculada levando em conta a aptidão de indivíduos de outras espécies. Essa abordagem foi estendida em Maruo & Delgado (2006) pela incorporação de um mecanismo auto-adaptativo no processo co-evolutivo hierárquico proposto anteriormente.

Um algoritmo co-evolutivo cooperativo multi-objetivos chamado MOCCA é proposto em Zong-Yi *et al.* (2006) para construir sistemas nebulosos de classificação,

levando em conta precisão e interpretabilidade. O algoritmo proposto inclui as fases de seleção de atributos, geração dos antecedentes das regras por meio de agrupamento nebuloso, geração do conseqüente das regras e finalmente a aplicação do AG co-evolutivo. Esse algoritmo co-evolutivo é composto de três espécies com a finalidade de otimizar a estrutura e parâmetros do sistema nebuloso gerado nas etapas anteriores.

O trabalho apresentado em Sim *et al.* (2004) baseia-se na abordagem co-evolutiva hospedeiro-parasita. O algoritmo proposto é chamado de algoritmo co-evolutivo *schema* e possui duas populações competitivas: a população hospedeira, formada por candidatos à solução e a população parasita, que representa um conjunto de esquemas que tem a finalidade de passar informação genética à população hospedeira no cálculo da função de aptidão.

O trabalho apresentado nesta dissertação é baseado na proposta de Delgado (Delgado *et al.* 2001; 2002; 2004), e enfoca a geração automática de sistemas nebulosos para classificação. A abordagem proposta aqui, bem como sua relação com o trabalho em que está fundamentado, será apresentada mais detalhadamente no próximo capítulo.

Capítulo 5

Sistema de Classificação Nebuloso Co-Evolutivo

Nesta seção será descrita a proposta co-evolutiva desenvolvida neste trabalho. Esta proposta baseia-se no trabalho de Delgado (Delgado, 2002; Delgado *et al.*, 2001; 2002; 2004), sendo que as principais diferenças entre o modelo apresentado aqui e o de Delgado estão no foco do trabalho e no cálculo das funções de aptidão. O sistema apresentado neste trabalho de dissertação visa à classificação de padrões através de um classificador nebuloso, enquanto que o sistema proposto por Delgado (2002) focaliza o problema de aproximação de funções via modelos de Takagi-Sugeno. Com relação ao cálculo das aptidões, aqui são considerados parâmetros adicionais, com o objetivo de estimular o surgimento de bases de regras mais compactas e regras nebulosas mais simples durante o processo evolutivo.

Segundo Delgado (2002), quando decisões 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 de várias populações.

A utilização de somente um indivíduo para codificar o sistema nebuloso completo (base de conhecimento, mecanismo de inferência, *t-norma*) pode se tornar inadequada porque o tamanho do espaço de busca será muito grande. Assim o ideal é dividir o problema em subproblemas de complexidades menores, tratando cada parte em subpopulações específicas: população de regras individuais, população de base de regras, população de partições nebulosas, etc.

A grande vantagem desta proposta está na determinação de todos os parâmetros do sistema nebuloso. A maioria das abordagens anteriores de sistemas nebulosos genéticos (seção 4.2) determinava somente uma parte da base de conhecimento (BRN ou BDN) ou as duas partes em processos distintos. A abordagem apresentada nesse trabalho, além de determinar a base de conhecimento, determina os outros parâmetros do sistema nebuloso como mecanismo de inferência e *t-norma* utilizada no cálculo do grau de compatibilidade de um padrão com uma regra nebulosa (seção 2.5.1).

A abordagem apresentada nessa dissertação se baseia no esquema de evolução hierárquica como descrito em Delgado (Delgado, 2002; Delgado *et al.*, 2001; 2002; 2004). 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 desempenho de um indivíduo dependa do desempenho de indivíduos pertencentes a outros módulos. Por meio destas relações cooperativas envolvendo indivíduos de diferentes espécies, os componentes dos sistemas nebulosos (representados por indivíduos de diferentes populações) são evoluídos.

5.1. Abordagem Co-Evolutiva do Classificador Nebuloso

A abordagem co-evolutiva supõe populações distintas, organizadas de forma hierárquica em módulos evolutivos. Indivíduos de diferentes populações, representando quatro espécies distintas, codificam diferentes componentes dos sistemas nebulosos. A abordagem considera:

- Uma população de partições nebulosas no primeiro nível (I);
- Uma população de regras nebulosas no segundo nível (II);
- Uma população de bases de regras no terceiro nível (III);
- Uma população de sistemas nebulosos no quarto nível (IV).

Como as populações são geneticamente isoladas, cada uma está associada a um algoritmo genético que é responsável pela evolução da população (figura 5.1). Porém, como todas as populações pertencem ao mesmo meio, as aptidões dos indivíduos são determinadas ao mesmo tempo, de acordo com o domínio da aplicação. Ou seja, os

indivíduos se cruzam somente com indivíduos da mesma população, mas interferem no desempenho dos indivíduos das outras populações.

A figura 5.1 ilustra as quatro populações isoladas no ambiente, tal que cada uma está associada a um algoritmo genético responsável pela evolução de sua respectiva população. Os quatro algoritmos genéticos são diferentes, ou seja, no ambiente existe um algoritmo genético para cada população. Isso é necessário porque indivíduos de uma população não podem se reproduzir e gerar descendentes com indivíduos de outra população. Porém os quatro algoritmos genéticos são relacionados um aos outros no cálculo das aptidões (detalhes na seção 5.4), ou seja, eles são dependentes hierarquicamente um do outro.

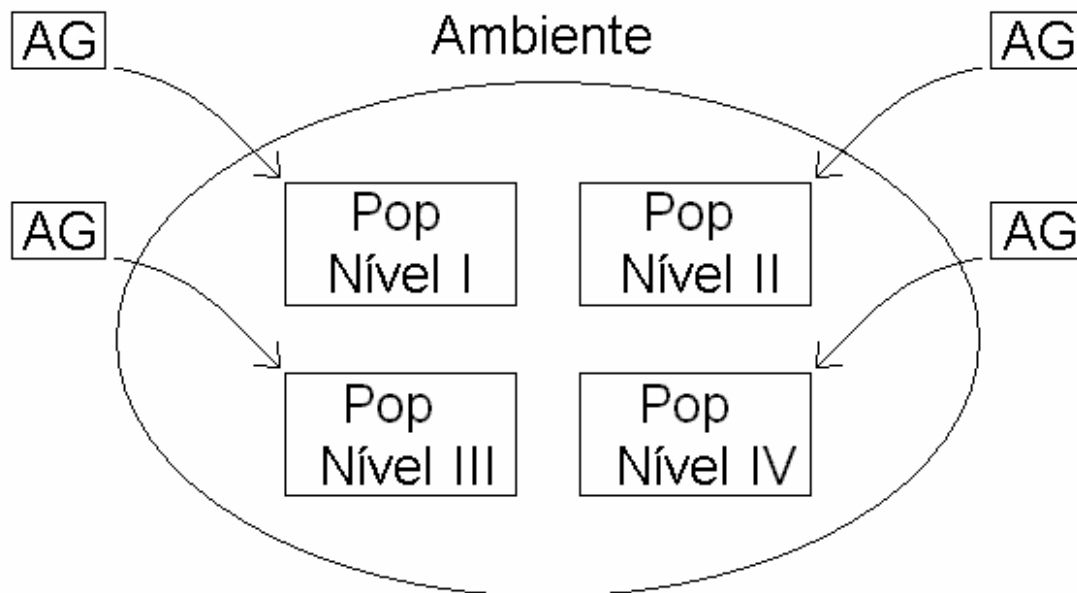


Figura 5.1: Populações geneticamente isoladas.

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 construção de indivíduos de outros níveis. A figura 5.2 ilustra o modo como foram estruturadas as populações nos seus níveis hierárquicos. Porém, para os cálculos das aptidões dos indivíduos das populações, essa hierarquia é invertida, sendo que as aptidões dos indivíduos dos níveis mais baixos dependem das aptidões dos indivíduos dos níveis mais altos (figura 5.2).

A figura 5.2 ilustra as quatro populações e a relação hierárquica entre elas, tanto nas relações de colaboração na geração das populações, como no cálculo das aptidões

dos indivíduos das populações. A colaboração acontece das populações dos níveis inferiores para as populações dos níveis superiores (detalhes na seção 5.3) e o cálculo das aptidões é feito primeiramente para as populações dos níveis superiores, e em seguida para as populações dos níveis inferiores (detalhes na seção 5.4).

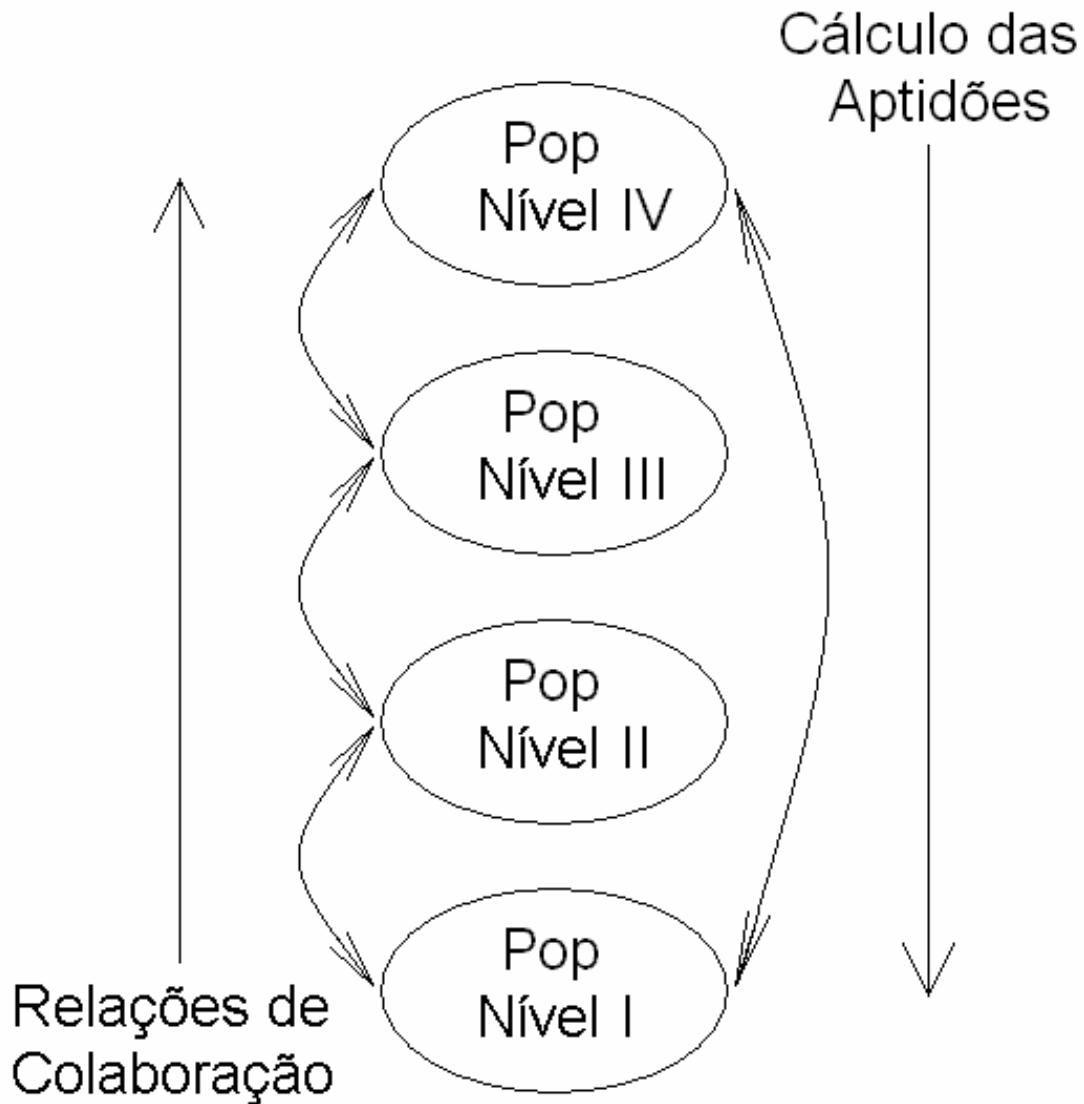


Figura 5.2: Hierarquia entre as relações de colaboração e entre os cálculos das adaptações.

5.2. Codificação Genética

Nesta seção será apresentado como foi representada a codificação genética de cada cromossomo dos indivíduos para cada uma das quatro populações presentes no

ambiente. Primeiramente é apresentada a codificação genética para os indivíduos da população de nível IV, e em seguida para os indivíduos da população de nível III, II e I.

Indivíduos da População IV

Os cromossomos que representam os indivíduos da população IV apresentam quatro genes. Eles estão organizados da seguinte forma:

- *Locus 1*: O alelo do gene presente no *locus 1* pode assumir um valor inteiro de 1 a 3, que representa a t-norma utilizada para encontrar o grau de compatibilidade entre o antecedente de uma regra e um padrão na aplicação do método de raciocínio;

As t-normas utilizadas são: (1) Intersecção Padrão, (2) Produto Algébrico, e (3) Diferença Limitada (detalhes na seção 2.2.2). A t-norma presente no *locus 1* é utilizada para encontrar o grau de compatibilidade entre uma regra e um padrão, que é o primeiro passo dos métodos de raciocínio nebuloso para classificação descritos na seção 2.5.1. O sistema foi projetado de forma modular, sendo de fácil alteração, para a remoção e/ou inserção de novas t-normas. Foram escolhidas as principais (mais usadas) somente por motivos práticos e pelo fato do foco principal do trabalho ser na geração automática de sistemas nebulosos e não no estudo das t-normas mais adequadas para a geração automática de sistemas nebulosos.

- *Locus 2*: O alelo do gene presente no *locus 2* pode assumir o valor inteiro 1 ou o 2, que representa o método utilizado no mecanismo de inferência para classificação;

Os métodos utilizados para classificação são: (1) Método de Raciocínio Nebuloso Clássico, e (2) Método de Raciocínio Nebuloso Geral (detalhes na seção 2.5.1). O sistema foi projetado de forma modular, sendo de fácil alteração, remoção e/ou inserção de novos métodos para o mecanismo de inferência. Porém isso não será necessário pelo fato do sistema já utilizar os principais métodos de inferência para classificação de padrões presentes até o momento na literatura.

- *Locus 3*: O alelo do gene presente no *locus 3* pode assumir um valor inteiro que associa o sistema nebuloso a um índice de uma base de regras da população do nível III;

- *Locus 4*: O alelo do gene presente no *locus 4* pode assumir um valor inteiro que associa o sistema nebuloso a um índice de uma partição nebulosa da população do nível I.

Por exemplo, o cromossomo da figura 5.3 representa um sistema nebuloso que utiliza a intersecção padrão como *t-norma*, o método de raciocínio nebuloso geral para classificação, e é composto pela base de regras do indivíduo de índice 5 da população do nível III e pela partição nebulosa de índice 4 da população do nível I.

1	2	5	4
---	---	---	---

Figura 5.3: Exemplo de cromossomo de um indivíduo da população do nível IV.

Indivíduos da População III

Os cromossomos que representam os indivíduos da população III são formados por índices dos indivíduos (regras nebulosas) da população II. Se o valor do índice for 0 (zero), representa nenhuma regra nebulosa, possibilitando bases de regras compactas.

O tamanho dos cromossomos dos indivíduos dessa população é igual a quantidade máxima de regras nebulosas que uma base de regras pode ter.

Por exemplo, o cromossomo da figura 5.4 representa um indivíduo da população do nível III que pode ter no máximo 5 regras nebulosas. Essa base de regras é formada pelas regras nebulosas de índices 9, 5, e 2 da população do nível II, e os *loci* 1 e 4 representam nenhuma regra, o que torna esse indivíduo da população do nível III mais compacto por possuir 3 regras nebulosas em sua base de regras e não 5 (quantidade máxima de regras nebulosas).

0	9	5	0	2
---	---	---	---	---

Figura 5.4: Exemplo de cromossomo de um indivíduo da população do nível III.

Durante o processo evolutivo é feita uma verificação para evitar que apareçam regras repetidas na mesma base de regras. Quando isso ocorre, somente uma regra nebulosa é considerada, e nas outras posições em que a mesma regra aparece, é atribuído índice 0, deixando a base de regras mais compacta.

É possível aparecer uma base de regras vazia, ou seja, um indivíduo da população cujo todos os índices são 0 (zero). Quando isso ocorre, esse indivíduo recebe valor de aptidão 0 (zero), conseqüentemente será excluído da próxima geração.

No decorrer do processo evolutivo, é possível também a presença de bases de regras duplicadas (redundância), embora seus respectivos cromossomos sejam diferentes. Por exemplo, o indivíduo da figura 5.4 formado pelos índices 0, 9, 5, 0 e 2, e um outro indivíduo formado pelos índices 9,5, 2, 0 e 0 aparecerem na mesma base de regras, teremos duas codificações diferentes para o mesmo conjunto de regras. Esses indivíduos não são eliminados, mas a diferenciação entre eles será feita no processo do cálculo do valor das aptidões, onde os valores de ambos provavelmente serão diferentes, já que essas bases de regras pertencerão a sistemas nebulosos diferentes ou a nenhum sistema nebuloso.

Indivíduos da População II

Os cromossomos que representam os indivíduos da população II possuem N genes, sendo que N é determinado por (*QUANTIDADE DE VARIÁVEIS LINGÜÍSTICAS* + 1). Por exemplo, se os padrões possuem 3 atributos (3 variáveis lingüísticas), mais sua respectiva classe, precisa-se de 4 (N) genes para armazenar a regra nebulosa. Os *loci* de 1 a ($N - 1$) representam os índices de cada termo lingüístico referentes a cada variável lingüística de uma partição nebulosa da população do nível I, e o *locus* N representa uma classe dentro do conjunto de classes (detalhes na seção 2.4.2).

Indivíduos da População I

O esquema de codificação adotado se baseia no conceito de posições relativas apresentado por Lee & Takagi (1993b) e no esquema adotado por Delgado (2002). Uma vez que o cromossomo é definido, um processo de decodificação é necessário para se gerar as funções de pertinência (Delgado *et al.*, 2001).

A escolha deste tipo de codificação tem por base em dois argumentos:

- Uniformidade: dois tipos de função de pertinência são utilizados neste trabalho (trapezoidal e triangular) e cada tipo necessita de um total de parâmetros diferentes. Portanto, optou-se por um esquema uniforme de

codificação baseado em cinco genes que, associado a um processo de decodificação, possibilitou a implementação destes dois tipos, com possibilidade de inclusão futura de outros tipos de funções;

- Relatividade: com o objetivo de reduzir o total de indivíduos inválidos resultantes da operação de cruzamento (recombinação genética), optou-se pelo esquema de codificação relativo que será apresentado a seguir.

Cada função de pertinência é representada por uma seqüência de cinco genes $\{S_k, L_k, C_{1k}, C_{2k}, R_k\}$, onde k identifica a posição da função no universo associado. Ou seja, a primeira função de pertinência (conjunto nebuloso) de uma variável lingüística (universo associado) é representada por $\{S_1, L_1, C_{11}, C_{21}, R_1\}$, onde $k = 1$. Para a segunda função de pertinência da mesma variável lingüística temos $k = 2$, e assim por diante.

As funções de pertinência utilizadas foram de dois tipos diferentes: (1) trapezoidal e (2) triangular (detalhes na seção 2.1.1). O tipo de função é definido pelo gene (S_k), localizado no *locus* 1 da seqüência de cinco genes. Valores nulos no *locus* 1 ($S_k = 0$) identificam que aquele termo lingüístico não fará parte da partição nebulosa da variável lingüística em questão. Os outros genes assumem valores relativos entre os parâmetros da função de pertinência k e da função de pertinência $k - 1$ (no caso onde $k = 1$ os valores relativos são obtidos com relação ao limite esquerdo do universo). No processo de decodificação assume-se que as funções trapezoidais e triangulares são representadas por 4 e 3 parâmetros respectivamente, conforme descrito na seção 2.1.1. Esses parâmetros são denotados aqui por $\{b_{1k}, m_{1k}, m_{2k}, b_{2k}\}$ para as funções trapezoidais e $\{b_{1k}, m_k, b_{2k}\}$ para as funções triangulares, como pode ser visto na figura 5.5.

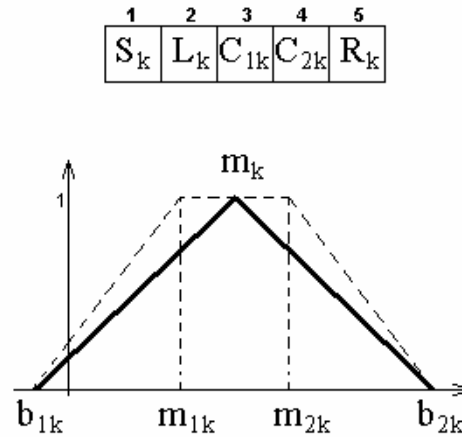


Figura 5.5: Esquema de codificação das funções de pertinência.

Os demais valores relativos que aparecem na codificação das funções de pertinência de um mesmo universo de discurso são facilmente compreendidos analisando-se a figura 5.6. No exemplo da figura 5.6 temos $S_1 = 1$, ou seja, a primeira função de pertinência é do tipo trapezoidal e $S_2 = 2$, ou seja, a segunda função de pertinência é do tipo triangular.

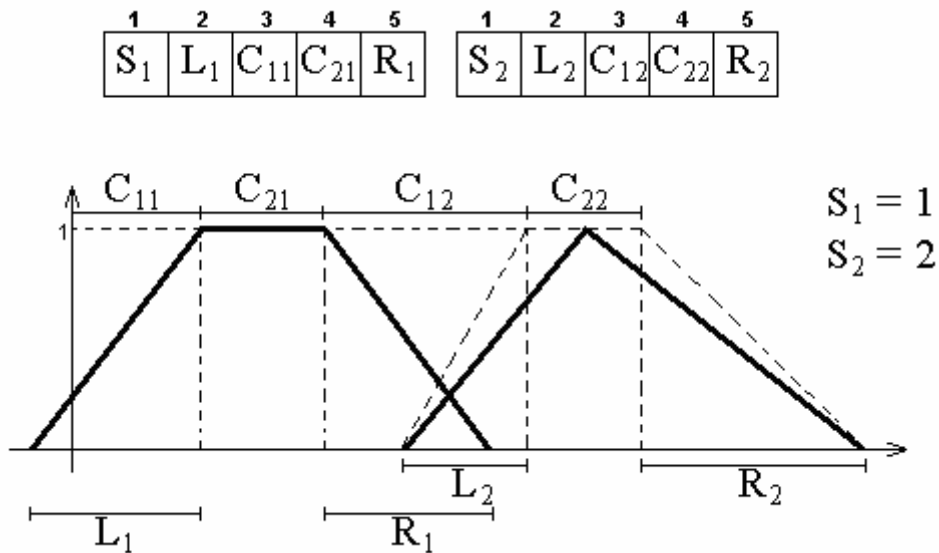


Figura 5.6: Posição relativa das funções de pertinência.

Logo, a posição absoluta de um termo específico no universo depende da posição do termo imediatamente anterior (figura 5.6). Para a k -ésima função de pertinência, os seus valores absolutos seriam dados por:

- $m_{1k} = m_{2k-1} + C_{1k}$;
- $m_{2k} = m_{1k} + C_{2k}$;
- $b_{1k} = m_{1k} - L_k$;
- $b_{2k} = m_{2k} + R_k$.

A figura 5.6 ilustra como se dá a posição relativa entre dois termos lingüísticos no universo.

Para a obtenção das funções do tipo trapezoidal, os genes dos *loci* 2 ao 5 são convertidos em seus valores absolutos $\{b_{1k}, m_{1k}, m_{2k}, b_{2k}\}$ (detalhes na seção 2.1.1) que definem os parâmetros $\{a, m, n, b\}$ destas funções. No caso de funções triangulares, após a determinação dos valores absolutos, o centro m_k (detalhes na seção 2.1.1) é encontrado pela média entre m_{1k} e m_{2k} (figura 5.5).

De acordo com a proposta apresentada (Lee & Takagi, 1993b; Delgado, 2002), cada indivíduo da população I apresenta (*QUANTIDADE DE VARIÁVEIS LINGÜÍSTICAS * QUANTIDADE DE VALORES LINGÜÍSTICOS * 5*) genes. Ou seja, se cada padrão apresenta 3 atributos, sendo que, cada atributo pode ter no máximo 2 termos lingüísticos, e que cada termo lingüístico precisa de 5 genes para ser representado, o cromossomo terá 30 genes.

5.3. Relações de Colaboração

O esquema de colaboração emerge das relações de interdependência entre indivíduos de diferentes espécies ou módulos evolutivos como podemos observar na figura 5.2 e na figura 5.7, onde as relações de colaboração na formação dos indivíduos partem dos níveis mais baixos para os níveis mais altos.

A figura 5.7 mostra os detalhes da hierarquia e colaboração que guiam o processo co-evolutivo para padrões com dois atributos. O *k-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. A 30ª base de regras do nível III contém, entre outras, a regra de número 15 que faz parte da população de regras nebulosas do nível II e é dada por:

$$\text{SE } (X_1 \text{ é "2"}) \text{ t-norma } (X_2 \text{ é "1"}) \text{ ENTÃO Classe} = C_i$$

Onde “2” e “1” identificam os termos lingüísticos definidos no 4º indivíduo do nível I, que é composto pelas funções de pertinência, e C_i identifica uma classe no conjunto de classes (detalhes na seção 2.5.1).

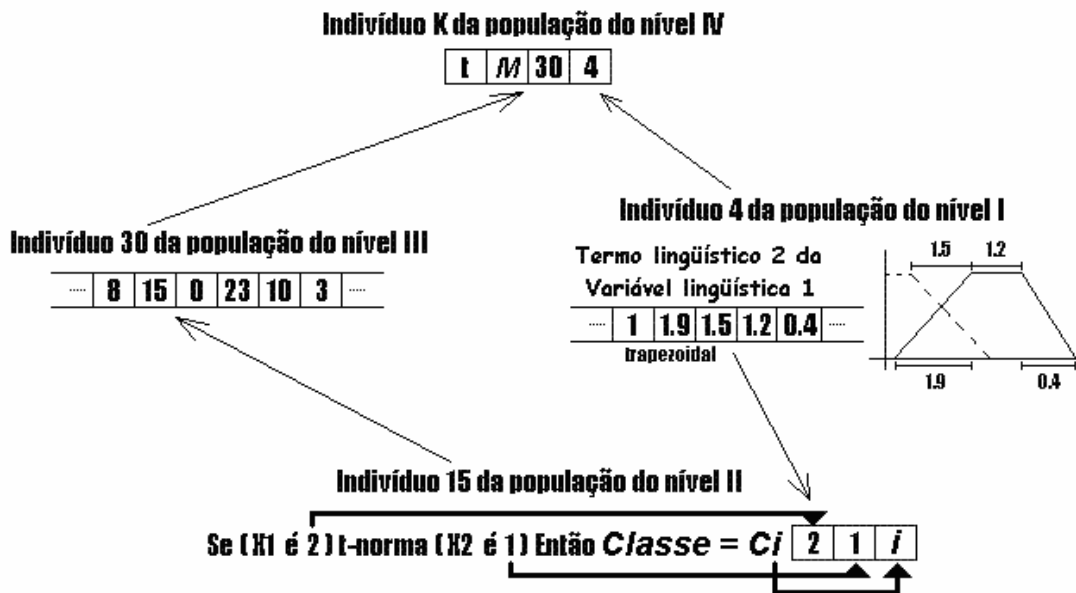


Figura 5.7: Relações de colaboração.

5.4. Cálculo das Aptidões

No processo de avaliação da aptidão de indivíduos pertencentes a diferentes espécies, relações intermodulares são estabelecidas: a aptidão de um indivíduo é calculada levando-se em conta a aptidão de indivíduos pertencentes a outras populações.

Pode-se observar na figura 5.2 que o processo de determinação da aptidão dos indivíduos no processo co-evolutivo segue a relação hierárquica entre as diferentes espécies, ou seja, ela se dá dos níveis mais altos para os níveis mais baixos.

A forma de avaliação dos indivíduos de cada população usada neste trabalho baseia-se na proposta apresentada em Delgado (2002), à qual foram feitas algumas adaptações. A avaliação é realizada da seguinte maneira:

- Sistema Nebuloso (Nível IV): o valor de aptidão de um sistema nebuloso SN , denotado por $F(SN)$ é baseado no desempenho do sistema nebuloso na aplicação, ou seja, é o desempenho do sistema nebuloso na solução do problema considerado em termos de número de classificações corretas;

- Base de Regras (Nível III): o valor de aptidão de uma base de regras BR_k é calculado por $F(BR_k) = \max(F(SN_m), \dots, F(SN_n)) + \alpha$, onde $F(SN_i)_{i=m, \dots, n}$ representam os valores de aptidões dos sistemas nebulosos que utilizam a base de regras k . O termo α representa inversamente a quantidade de regras nebulosas na base de regras k , ou seja, quanto menos regras nebulosas tiver a base de regras, melhor será a sua aptidão;
- Regras Nebulosas (Nível II): o valor de aptidão de uma regra RN_k é calculado por $F(RN_k) = \text{med}(F(BR_m), \dots, F(BR_n)) + \beta$, onde $F(BR_i)_{i=m, \dots, n}$ identificam os valores de aptidões das bases de regras que utilizam a regra nebulosa k ; med é o operador de média aritmética. O termo β representa a quantidade de condições irrelevantes na regra nebulosa k , ou seja, quanto mais condições irrelevantes a regra nebulosa tiver, melhor o indivíduo;
- Partição Nebulosa (Nível I): $F(PN_k) = \max(F(SN_m), \dots, F(SN_n))$, onde $F(SN_i)_{i=m, \dots, n}$ representam os sistemas nebulosos que utilizam a partição nebulosa k .

A aptidão de cada indivíduo dos níveis I e III é calculada com base 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 BRN e BDN , define o valor do desempenho de sua base de regras e partição nebulosa respectivamente. 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, sua avaliação não deve ser piorada devido à sua participação em um sistema nebuloso de desempenho baixo, daí a razão de se usar o operador de máximo. Para as regras nebulosas (Nível II), uma vez que existe uma grande interação delas e uma regra sozinha dificilmente responde pelo desempenho do sistema nebuloso, a média é mais indicada. Neste caso, a regra é avaliada pela média do quanto ela coopera com outras regras para compor um sistema nebuloso no nível IV.

As mudanças feitas no cálculo das aptidões, com relação à usada no trabalho em que esta proposta se baseia, têm como objetivo obter melhores sistemas nebulosos com relação à interpretação. Essas mudanças são apresentadas a seguir:

- Compactação: no cálculo da aptidão de cada base de regras BR_k foi acrescentada a constante α , que representa o tamanho da base de regras, isto é, a quantidade de regras nebulosas na base de regras;
- Simplicidade: no cálculo da aptidão de cada regra RN_k foi acrescentada a constante β , que representa a quantidade de condições irrelevantes, ou seja, o tamanho da regra nebulosa.

5.5. População Inicial

Nesta seção será apresentado como foram geradas as populações iniciais para cada nível. Primeiramente é apresentado como é a inicialização da população do nível I, e em seguida como é a inicialização das populações de nível II, III e IV.

População I

As partições nebulosas são inicializadas com conjuntos nebulosos uniformes nos respectivos universos. Os tipos de funções de pertinências desses conjuntos são definidas de forma aleatória.

População II

As regras nebulosas são inicializadas de forma aleatória tanto para o termo lingüístico de cada variável lingüística quanto para a classe. Na inicialização não existem regras genéricas (regras em que não aparecem todas as variáveis lingüísticas no antecedente), elas vão aparecendo no decorrer da evolução.

População III

As bases de regras são geradas com regras nebulosas aleatórias, com o máximo de regras nebulosas permitidas por bases de regras, ou seja, a base de regra é a maior possível em quantidade de regras nebulosas. Isso significa que base de regras compactas só aparecerá durante o processo de evolução. A única restrição é que não pode haver regras repetidas e/ou conflitantes na população inicial.

População IV

Os quatro componentes dos sistemas nebulosos são gerados aleatoriamente.

5.6. Método de Seleção

O método de seleção utilizado foi a seleção proporcional à aptidão (roleta). O processo de seleção é combinado com o elitismo a fim de não perder os melhores indivíduos nas próximas gerações (detalhes na seção 3.2.2). O sistema foi projetado de forma modular, sendo de fácil alteração e/ou inserção de novos métodos de seleção. Foi escolhido um método de fácil implementação somente por motivos práticos e pelo fato do foco ser na geração automática de sistemas nebulosos e não no estudo do método de seleção mais adequado para a geração automática de sistemas nebulosos.

5.7. Operadores Evolutivos

O cruzamento utilizado foi o de um ponto, onde um *locus* do cromossomo é escolhido aleatoriamente (detalhes na seção 3.2.3). A mutação utilizada foi a de alterar somente o valor de um *locus* escolhido aleatoriamente no cromossomo (detalhes na seção 3.2.3). O sistema está modular, sendo de fácil alteração e/ou inserção de novos métodos de cruzamento e/ou mutação. Foram escolhidos métodos de fácil implementação somente por motivos práticos e pelo fato do foco ser na geração automática de sistemas nebulosos e não no estudo do método de cruzamento e/ou mutação mais adequado para a geração automática de sistemas nebulosos.

Não foi feito nenhum tratamento para se evitar indivíduos inválidos ou repetidos no processo evolutivo (cruzamento e mutação). Há apenas um processo indireto de correção, no qual indivíduos repetidos são automaticamente eliminados, e indivíduos inválidos normalmente correspondem a sistemas nebulosos com desempenho baixo, indicando maior probabilidade de exclusão em gerações futuras.

5.8. Algoritmo Genético

O algoritmo genético utilizado para todas as populações do classificador nebuloso é apresentado na figura 5.8. São quatro AGs independentes (um para cada população) interligados pelo processo de cálculo das aptidões (detalhes na seção 5.1).

Algoritmo

rotina NOVA GERAÇÃO

início

seleciona melhores indivíduos (elitismo) = N ;

para *quantidade de indivíduos* – N (incremento de 2 em 2) **faça**

selecionar pais da população anterior;

cruzamento dos pais gerando os descendentes;

mutação genética nos descendentes;

fim para

fim

fim rotina

início

geração $\leftarrow 0$;

inicia todas as populações;

avalia todas as populações;

enquanto não desempenho aceitável e não quantidade máxima de gerações **faça**

geração \leftarrow geração + 1;

NOVA GERAÇÃO para todas as populações;

avalia todas as populações;

fim enquanto

fim

Figura 5.8: Algoritmo genético utilizado para gerar o classificador nebuloso.

No capítulo a seguir são apresentados os resultados das simulações feitas em vários domínios. São apresentadas as simulações e os resultados obtidos em aplicações no \mathcal{R}^2 , no \mathcal{R}^3 , e no \mathcal{R}^4 .

Capítulo 6

Simulações e Resultados

O objetivo deste capítulo é descrever os resultados dos testes envolvendo os sistemas nebulosos resultantes da abordagem co-evolutiva. Além da acuidade, será analisada a interpretabilidade (detalhes na seção 4.1) dos sistemas nebulosos obtidos. Fica clara a autonomia da proposta, já que somente os parâmetros da co-evolução são determinados por intervenção externa. Os parâmetros dos sistemas nebulosos são determinados durante o processo evolutivo.

Este capítulo traz os resultados aplicados em sistemas de classificação. Os testes representam: classificação no \mathcal{R}^2 , utilizando os dados das espirais concêntricas (apêndice A); classificação no \mathcal{R}^4 , utilizando os dados da base de dados IRIS (dados extraídos do repositório UCI (Merz & Murphy, 1998)); e, classificação no \mathcal{R}^3 , utilizando os dados da base de dados AUTO-MPG (dados extraídos do repositório UCI (Merz & Murphy, 1998)).

A base de dados das espirais concêntricas e a base de dados IRIS foram escolhidas porque essas duas bases de dados foram utilizadas no trabalho de Delgado (2002), o que possibilita análises comparativas desta abordagem com a abordagem utilizada como base para essa dissertação. A base de dados AUTO-MPG foi escolhida por já ter sido usada em trabalhos de um projeto mais abrangente intitulado *Construção e Otimização de Base de Conhecimento Fuzzy (Uma Abordagem Evolutiva)* no qual essa dissertação está inserida.

6.1. Abordagens de Comparação

Nessa seção serão relatados os sistemas utilizados para comparação com o classificador nebuloso apresentado nesse trabalho.

- **ANFIS:** Sistema neuro-nebuloso utilizado na produção de sistemas híbridos (Jang, 1993).
- **CompacFuzzClass:** Classificador nebuloso proposto por Castellano & Fanelli (2000), denominado por Delgado (2002) de CompacFuzzyClass.
- **EGFS:** Sistema genético nebuloso evolutivo (*Evolutionary Genetic Fuzzy System*) que utiliza a abordagem evolutiva tradicional, implementado por Delgado (2002) para comparação com a abordagem co-evolutiva.
- **CoevolGFS:** Sistema genético nebuloso co-evolutivo (*Coevolutionary Genetic Fuzzy System*) que utiliza da abordagem co-evolutiva na produção de sistemas nebuloso (Delgado, 2002; Delgado *et al.*, 2001; 2002; 2004).

6.2. Considerações Preliminares

Inicialmente alguns termos utilizados para apresentação dos resultados e suas abreviações são definidos:

- **Quantidade de padrões (*PD*):** representa a quantidade de padrões presentes no conjunto de dados sendo tratado;
- **Quantidade de atributos (*AT*):** representa a quantidade de atributos que descrevem cada padrão no conjunto de dados;
- **Quantidade de classes (*CL*):** representa a quantidade de classes às quais os padrões podem ser associados;
- **Quantidade de padrões por classe (*PC*):** representa a quantidade de padrões por cada classe presentes no conjunto de dados;
- **Quantidade máxima de termos lingüísticos (*TL*):** representa a quantidade máxima de termos lingüísticos que as variáveis lingüísticas podem assumir e é fixo para todas as variáveis lingüísticas. Esse parâmetro é igual para todas as variáveis lingüísticas, o que não significa

que a quantidade de termos lingüísticos será a mesma para todas as variáveis lingüísticas no final do processo evolutivo;

- **Quantidade máxima de regras nebulosas na base de regras (RN):** representa a quantidade máxima de regras nebulosas que uma base de regras pode ter;
- **Tamanho das populações ($P_I, P_{II}, P_{III}, P_{IV}$):** as quatro populações têm seus tamanhos definidos por P_I (população do nível I), P_{II} (população do nível II), P_{III} (população do nível II) e P_{IV} (população do nível IV);
- **Número de gerações (GR):** representa o número máximo de gerações no processo evolutivo;
- **Porcentagem de elitismo (EL):** representa a porcentagem de elitismo na seleção no processo evolutivo;
- **Probabilidade de cruzamento (CZ):** representa a probabilidade de cruzamento entre dois indivíduos no processo evolutivo;
- **Probabilidade de mutação (MT):** representa a probabilidade de mutação de um indivíduo no processo evolutivo.

Como cada população tem um algoritmo genético que atua sobre ela, os parâmetros de número de gerações (GR), porcentagem de elitismo (EL), probabilidade de cruzamento (CZ) e probabilidade de mutação (MT) são independentes para cada algoritmo genético. É possível atribuir valores diferentes para cada um desses parâmetros para cada uma das populações.

As simulações apresentadas envolvem um número relativamente pequeno de variáveis lingüísticas, portanto a população do nível II não sofre alteração no processo evolutivo. O nível II é fixo e representado por um conjunto formado por todas as possíveis regras nebulosas. Assim, $P_{II} = [(TL + 1)^{AT} - 1] * CL$.

Para a fase de treinamento e para a fase de validação dos sistemas nebulosos gerados, o conjunto de dados é dividido em dois conjuntos: conjunto de treinamento e conjunto de teste. O termo porcentagem de treinamento (PT) que representa a porcentagem de padrões de cada classe que formará o conjunto de treinamento, o restante dos padrões formará o conjunto de teste. Dessa forma, o conjunto de dados é primeiramente separado em subconjuntos de acordo com suas classes. Em seguida, a porcentagem de treinamento (PT) é aplicada a cada um desses subconjuntos para

geração aleatória dos exemplos de treinamento, de acordo com a quantidade de padrões por classe.

Nas tabelas apresentadas com os resultados aparecem duas colunas que merecem uma atenção por terem significados diferentes para o trabalho: **ciclos** e **vezes**. A coluna **ciclos** representa a quantidade de ciclos ou gerações necessárias para se gerar um sistema nebuloso. A coluna **vezes** representa a quantidade de vezes que o programa foi executado, gerando cada uma um sistema nebuloso diferente. Para cada vez, o sistema nebuloso co-evolutivo é reinicializado, ou seja, as populações começam da primeira geração (isto significa indivíduos diferentes, já que as populações iniciais começam de forma aleatória como descrito na seção 5.5). A partir dos desempenhos dos sistemas nebulosos gerados em cada vez, foi possível determinar a média e o desvio padrão para cada uma das aplicações.

Para a comparação dos resultados com outros sistemas presentes na literatura, o Sistema Nebuloso Co-Evolutivo apresentado nesse trabalho será referenciado por SisCoE para efeito de simplificação. As próximas seções descrevem os resultados obtidos para as bases de dados utilizadas na validação do classificador nebuloso gerado automaticamente.

6.3. Espirais Concêntricas

O problema das espirais concêntricas é bastante conhecido na literatura. O problema consiste em classificar corretamente os pontos pertencentes a duas espirais concêntricas de origem no ponto $(0,0)$ do plano cartesiano (figura 6.1).

A base de dados é formada pelos pontos pertencentes as espirais (apêndice A). Uma espiral representa a classe +1 e a outra espiral representa a classe -1 do conjunto de dados (figura 6.1).

O principal objetivo do classificador é associar corretamente cada par (x_1, x_2) à sua espiral apropriada.

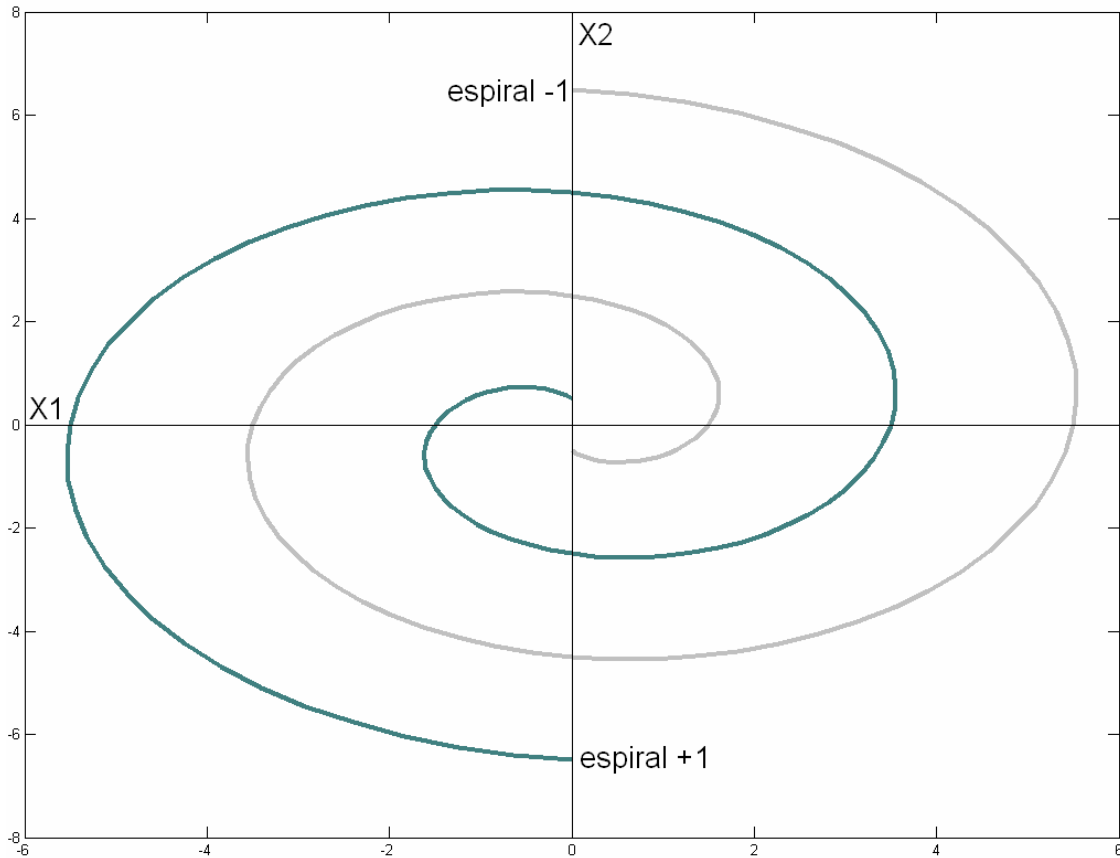


Figura 6.1: Espirais concêntricas.

A seguir são apresentados os parâmetros utilizados para essa aplicação. Os parâmetros são os mesmos apresentados em Delgado (2002).

Parâmetros do conjunto de exemplos:

- Quantidade de padrões $PD = 194$;
- Quantidade de atributos $AT = 2$;
- Quantidade de classes $CL = 2$;
- Quantidade de padrões por classe $PC = 97$.

Parâmetros dos sistemas nebulosos:

- Quantidade máxima de termos lingüísticos $TL = 3$;
- Quantidade máxima de regras nebulosas na base de regras $RN = 4$.

Parâmetros do algoritmo genético:

- Tamanho da população do nível I $P_I = 20$;

- Tamanho da população do nível II $P_{II} = 30$;
- Tamanho da população do nível III $P_{III} = 80$;
- Tamanho da população do nível IV $P_{IV} = 100$;
- Número de gerações $GR = 529$;
- Porcentagem de elitismo $EL = 5\%$;
- Probabilidade de cruzamento $CZ = 90\%$;
- Probabilidade de mutação $MT = 10\%$.

A porcentagem de treinamento PT utilizada foi igual a 100%, ou seja, os sistemas nebulosos gerados tentam classificar corretamente todo o conjunto de dados de treinamento, não foi utilizado conjunto de dados de teste.

A tabela 6.1 apresenta os resultados obtidos em Delgado (2002) para duas abordagens, o ANFIS e o CoevolGFS.

Tabela 6.1: Resultados do problema das espirais concêntricas.

Abordagem	Ciclos	Regras	Erros
ANFIS*	1000	9	18
CoevolGFS*	529	4	0

* resultados obtidos em (Delgado, 2002).

Os resultados obtidos para essa base de dados com os mesmos parâmetros utilizados por Delgado (2002) no CoevolGFS foram insatisfatórios para o SisCoE. Foram necessárias alterações nos parâmetros para obter-se resultados melhores. As alterações foram feitas na quantidade de regras, na quantidade de ciclos e na quantidade máxima de termos lingüísticos. O resultado obtido pelo SisCoE para a base de dados das espirais concêntricas pode ser observado na tabela 6.2.

Tabela 6.2: Resultados das simulações para o problema das espirais concêntricas.

Abordagem	Ciclos	Regras	TL	Erros
SisCoE	1000	7	5	19

O melhor sistema nebuloso gerado automaticamente através do SisCoE utiliza como mecanismo de inferência o raciocínio nebuloso geral, sendo o produto algébrico a

t-norma utilizada para encontrar o grau de compatibilidade entre um padrão e uma regra nebulosa.

O sistema nebuloso apresenta as partições nebulosas da figura 6.2 e a seguinte base de regras:

SE x_1 é T1 ENTÃO +I

SE x_1 é T2 ENTÃO +I

SE x_1 é T4 ENTÃO +I

SE x_2 é T2 ENTÃO -I

SE x_2 é T3 ENTÃO -I

SE x_2 é T4 ENTÃO -I

SE x_2 é T5 ENTÃO -I

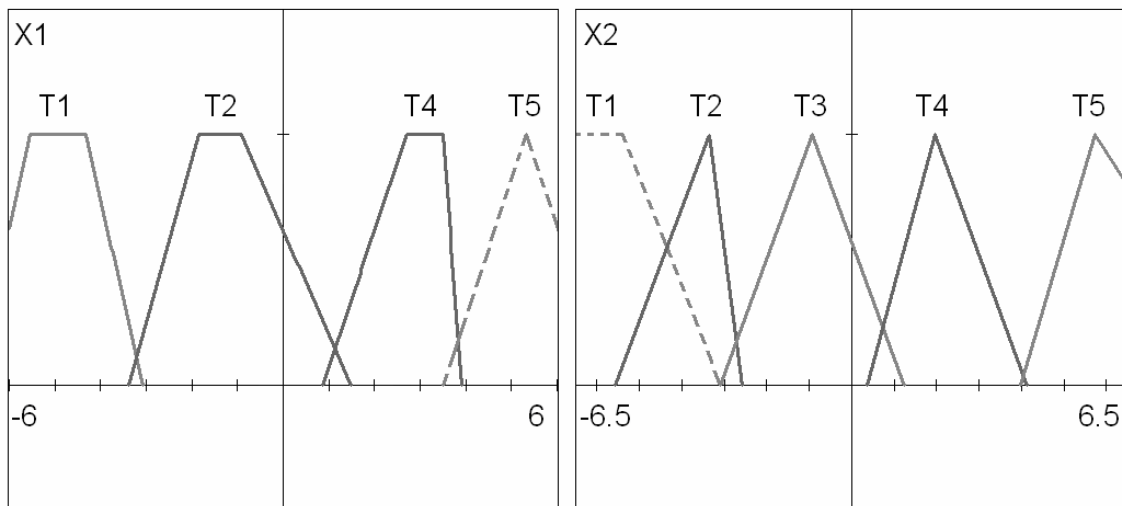


Figura 6.2: Partições nebulosas para o problema das espirais concêntricas.

A partir da base de regras e das partições nebulosas (figura 6.2) podemos observar que o termo lingüístico T5 da variável lingüística x_1 e o termo lingüístico T1 da variável lingüística x_2 não aparece em nenhuma regra nebulosa. Isso não significa que o domínio das variáveis lingüísticas x_1 e x_2 não estão cobertos, porque a base de regras apresenta regras genéricas, ou seja, regras com apenas uma variável no antecedente. Assim, é possível determinar a classe a que pertence um padrão com base no valor de apenas um dos atributos, independente do valor do outro atributo.

A tabela 6.3 apresenta os resultados de média e desvio padrão para a base de dados das espirais concêntricas.

Tabela 6.3: Média e desvio padrão para o problema das espirais concêntricas.

Abordagem	Vezes	Média	Desvio Padrão
SisCoE	10	90.6650%	3.2232

6.4. IRIS

Esse problema de classificação envolve três espécies (classes) de flores (*Iris setosa*, *Iris versicolor* e *Iris virginica*). Os padrões possuem quatro atributos, sendo esses: comprimento da sépala da flor (x_1); largura da sépala da flor (x_2); comprimento da pétala da flor (x_3); e, largura da pétala da flor (x_4).

A seguir são apresentados os parâmetros utilizados para essa aplicação. Os parâmetros são os mesmos apresentados em Delgado (2002). A intenção é que a comparação entre as abordagens seja a mais fiel possível.

Parâmetros do conjunto de exemplos:

- Quantidade de padrões $PD = 150$;
- Quantidade de atributos $AT = 4$;
- Quantidade de classes $CL = 3$;
- Quantidade de padrões por classe $PC = 50$.

Parâmetros dos sistemas nebulosos:

- Quantidade máxima de termos lingüísticos $TL = 2$;
- Quantidade máxima de regras nebulosas na base de regras $RN = 4$.

Parâmetros do algoritmo genético:

- Tamanho da população do nível I $P_I = 10$;
- Tamanho da população do nível II $P_{II} = 240$;
- Tamanho da população do nível III $P_{III} = 70$;
- Tamanho da população do nível IV $P_{IV} = 30$;

- Número de gerações $GR = 100$;
- Porcentagem de elitismo $EL = 5\%$;
- Probabilidade de cruzamento $CZ = 90\%$;
- Probabilidade de mutação $MT = 10\%$.

A porcentagem de treinamento PT utilizada foi igual a 50%, ou seja, metade dos dados de cada classe (25 exemplos) foram usados para o treinamento e a outra metade para o teste dos sistemas nebulosos gerados.

A tabela 6.4 apresenta o resultado obtido pelo SisCoE em relação a outros sistemas da literatura para o conjunto de dados IRIS e a tabela 6.5 apresenta os resultados de média e desvio padrão para essa aplicação.

Tabela 6.4: Resultados das simulações para o conjunto de dados IRIS.

Abordagem	Ciclos	Regras	Erros
CompacFuzzClass*	---	5	4
ANFIS*	100	16	6
EGFS*	100	13	8
CoevolGFS*	100	4	3
SisCoE	100	4	2

* resultados obtidos em (Delgado, 2002).

Tabela 6.5: Média e desvio padrão para o conjunto de dados IRIS.

Abordagem	Vezes	Média	Desvio Padrão
SisCoE	50	94.2888%	1.1006

O melhor sistema nebuloso gerado automaticamente através do SisCoE utiliza como mecanismo de inferência o raciocínio nebuloso clássico, sendo a intersecção padrão a *t-norma* utilizada para encontrar o grau de compatibilidade entre um padrão e uma regra nebulosa.

O sistema nebuloso apresenta as partições nebulosas da figura 6.3 e a seguinte base de regras:

SE x_1 é BAIXO **E** x_2 é BAIXO **E** x_4 é BAIXO **ENTÃO** *Iris versicolor*

SE x_1 é BAIXO **E** x_3 é BAIXO **E** x_4 é BAIXO **ENTÃO** *Iris setosa*

SE x_1 é ALTO **E** x_2 é ALTO **E** x_3 é BAIXO **ENTÃO** *Iris virginica*

SE x_1 é ALTO **E** x_3 é ALTO **ENTÃO** *Iris virginica*

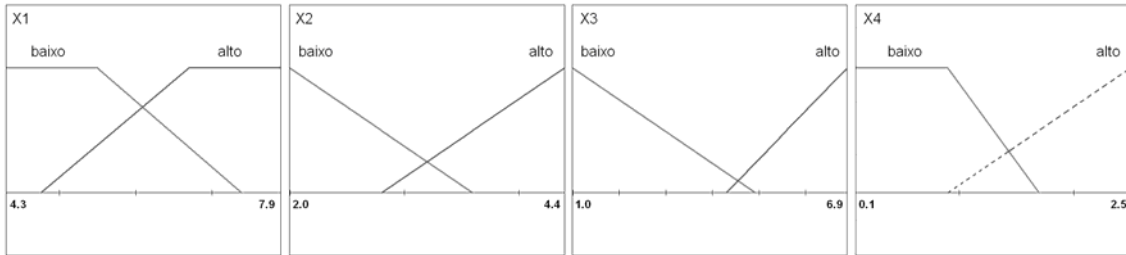


Figura 6.3: Partições nebulosas para o problema da IRIS.

A partir da base de regras e das partições nebulosas (figura 6.3) podemos observar que o termo lingüístico ALTO da variável lingüística x_4 não aparece em nenhuma regra nebulosa. Isso não significa que o domínio da variável lingüística x_4 não está coberto, porque a base de regras apresenta regras genéricas, ou seja, regras que não determinam valores para todos os atributos. Como a terceira e quarta regras da base não estipulam valores para a variável x_4 , é possível determinar a classe a que pertence um padrão com base nos valores das outras três variáveis, independente do valor de x_4 .

Pode-se observar a superioridade das abordagens co-evolutivas (SisCoE e CoevolGFS) sobre as outras através da tabela 6.4. Podemos observar também que o SisCoE apresentou menor quantidade de erros com relação ao CoevolGFS em um sistema tipicamente de classificação.

Foram feitos testes mudando a quantidade máxima de termos lingüísticos TL para 3. A intenção era tentar aumentar a acuidade do sistema para essa base de dados, o que não ocorreu, a acuidade do sistema se manteve a mesma (mesma quantidade de erros) e a média não diminuiu. A única desvantagem foi o tempo médio para a geração de um sistema nebuloso, que aumentou cerca de 40% (de ~25 segundo para ~35 segundos). Por não apresentar um desempenho pior, o sistema nebuloso co-evolutivo mostrou-se pouco sensível a este parâmetro. O resultado da simulação é apresentado na tabela 6.6, e os resultados da média e do desvio padrão são apresentados na tabela 6.7.

Tabela 6.6: Resultados das simulações para o conjunto de dados IRIS (3 termos lingüísticos).

Abordagem	Ciclos	Regras	Erros
SisCoE	100	4	2

Tabela 6.7: Média e desvio padrão para o conjunto de dados IRIS (3 termos lingüísticos).

Abordagem	Vezes	Média	Desvio Padrão
SisCoE	50	94.1298%	1.4104

6.5. AUTO-MPG

O conjunto de dados AUTO-MPG original foi transformado e reduzido para ser utilizado neste trabalho. O conjunto de dados original possui 8 atributos de carros de diversas marcas e modelos, e uma classe contínua, que indica o consumo. Três desses atributos foram selecionados para representar um padrão: cavalos de força (x_1); peso (x_2); e, aceleração (x_3). Esses três atributos foram escolhidos porque trabalhos anteriores do projeto intitulado *Construção e Otimização de Base de Conhecimento Fuzzy (Uma Abordagem Evolutiva)* utilizaram esses mesmos atributos. Os padrões são classificados de acordo com o consumo (mpg - miles per gallon). Como essa classe é originalmente contínua (pode assumir valores reais), foi necessário transformá-la para classe discreta. Essa transformação aconteceu dividindo o universo de valores possíveis para consumo em três intervalos consecutivos, que passaram a representar as classes de consumo baixo, médio e alto.

A seguir são apresentados os parâmetros utilizados para essa aplicação. Os parâmetros foram determinados empiricamente para esse conjunto de dados.

Parâmetros do conjunto de exemplos:

- Quantidade de padrões $PD = 392$;
- Quantidade de atributos $AT = 3$;
- Quantidade de classes $CL = 3$;
- Quantidade de padrões por classe $PC = 131$ para as classes baixo e alto, e 130 para a classe médio.

Parâmetros dos sistemas nebulosos:

- Quantidade máxima de termos lingüísticos $TL = 3$ para a primeira simulação e 5 para a segunda simulação;
- Quantidade máxima de regras nebulosas na base de regras $RN = 6$.

Parâmetros do algoritmo genético:

- Tamanho da população do nível I $P_I = 20$;
- Tamanho da população do nível II $P_{II} = 189$ para 3 termos lingüísticos e 645 para 5 termos lingüísticos;
- Tamanho da população do nível III $P_{III} = 100$;
- Tamanho da população do nível IV $P_{IV} = 50$;
- Número de gerações $GR = 500$;
- Porcentagem de elitismo $EL = 5\%$;
- Probabilidade de cruzamento $CZ = 90\%$;
- Probabilidade de mutação $MT = 10\%$.

A porcentagem de treinamento PT utilizada foi igual a 75%, ou seja, $\frac{3}{4}$ dos dados de cada classe foram usados para o treinamento e o restante para o teste dos sistemas nebulosos gerados.

A tabela 6.8 apresenta os resultados obtidos pelo SisCoE nas simulações com 3 e com 5 termos lingüísticos para o conjunto de dados AUTO-MPG e a tabela 6.9 apresenta os resultados de média e desvio padrão para essa aplicação.

Tabela 6.8: Resultados das simulações para o conjunto de dados AUTO-MPG.

Abordagem	TL	Regras	Erros	Desempenho
SisCoE	3	5	14	85.8586%
SisCoE	5	5	15	84.8485%

Tabela 6.9: Média e desvio padrão para o conjunto de dados AUTO-MPG.

Abordagem	TL	Veze	Média	Desvio Padrão
SisCoE	3	50	82.9614%	0.9035
SisCoE	5	50	82.9210%	0.8142

Capítulo 7

Considerações Finais

Este trabalho apresentou um sistema para geração automática de sistemas nebulosos. Esse sistema é baseado na co-evolução, onde a hierarquia e colaboração entre várias populações de um mesmo meio ambiente têm um papel importante no processo evolutivo.

As simulações feitas no **Sistema Nebuloso Co-Evolutivo (SisCoE)** mostraram autonomia e flexibilidade para encontrar modelos acurados. Os resultados obtidos confirmam a expectativa de que a co-evolução, associada à hierarquia e co-operação, representa uma alternativa promissora para o projeto automático de sistemas nebulosos. Os resultados iniciais são promissores e pode-se observar através deles que os sistemas nebulosos gerados apresentam algumas características relevantes que merecem ser destacadas:

- Conjuntos nebulosos bem definidos para as variáveis lingüísticas, o que possibilita boa visibilidade;
- Regras genéricas, o que significa boa interpretação humana já que não aparecem todas variáveis lingüísticas em todas as regras;
- Base de regras compacta, com poucas regras;
- Base de regras consistente. A consistência é garantida pela ausência de regras redundantes: se aparecer mais de uma regra igual no processo evolutivo, apenas uma é mantida na base de regras e as outras são eliminadas. E no caso de aparecerem regras inconsistentes, por exemplo, mesmo antecedentes e conseqüentes diferentes, a função de cálculo das

aptidões normalmente dá um valor de aptidão muito baixo para essa base de regras, o que significa uma alta probabilidade de exclusão em gerações futuras.

Porém, alguns problemas foram encontrados. Para base de dados muito grandes, com várias variáveis, surgiram problemas com limite de memória gerenciável da plataforma usada (JAVA). Quando esse problema foi resolvido, mesmo colocando toda memória RAM (512 MB) disponível no processamento, o sistema ficou muito lento, o que tornou inviável sua aplicação a grandes bases de dados.

7.1. Trabalhos Futuros

Estão previstos para trabalhos futuros:

- Investigação de melhorias no sistema como de mecanismos otimizados de implementação;
- Investigação da possível utilização de sistemas paralelos e/ou distribuídos como uma possível solução do tamanho da memória RAM;
- Investigação do desempenho do sistema para grandes bases de dados;
- Utilização de codificação binária nos cromossomos dos indivíduos da população de nível III para verificar se o sistema apresenta melhorias;
- Investigação de outras funções de avaliação do desempenho dos sistemas nebulosos gerados, por exemplo, não verificar só o desempenho do sistema nebuloso com relação a quantidade de erros, e sim verificar o desempenho com relação a falsos positivos e falsos negativos;
- Verificação do desempenho sem as penalizações α e β (detalhes na seção 5.4). Podem ser obtidos melhores desempenhos sem levar em conta base de regras compactas e regras simples;
- Verificação da utilização de seleção por torneio no processo de seleção dos indivíduos mais aptos das populações;
- Utilização de validação cruzada para determinar a acuidade dos sistemas nebulosos gerados automaticamente;

- Utilização de algoritmos genéticos auto-adaptativos visando ganho de desempenho em termos de tempo e/ou acuidade dos sistemas nebulosos gerados automaticamente.

Apêndice A

Base de dados de duas espirais concêntricas. Os padrões são compostos por $(x_1, x_2, \text{espiral})$.

0.00000,-6.50000,+1	-2.26859,4.24424,+1	2.59834,-1.73615,+1	-1.43057,0.14089,+1
-0.63098,-6.40650,+1	-1.81774,4.38842,+1	2.36734,-1.94282,+1	-1.34857,0.26824,+1
-1.24370,-6.25250,+1	-1.36070,4.48565,+1	2.12132,-2.12132,+1	-1.25598,0.38099,+1
-1.83242,-6.04068,+1	-0.90229,4.53613,+1	1.86353,-2.27071,+1	-1.15484,0.47835,+1
-2.39177,-5.77424,+1	-0.44720,4.54053,+1	1.59726,-2.39047,+1	-1.04728,0.55978,+1
-2.91676,-5.45688,+1	-0.00000,4.50000,+1	1.32580,-2.48040,+1	-0.93540,0.62501,+1
-3.40286,-5.09275,+1	0.43495,4.41613,+1	1.05237,-2.54066,+1	-0.82132,0.67404,+1
-3.84600,-4.68637,+1	0.85352,4.29093,+1	0.78014,-2.57177,+1	-0.70710,0.70710,+1
-4.24264,-4.24264,+1	1.25185,4.12680,+1	0.51211,-2.57456,+1	-0.59474,0.72469,+1
-4.58974,-3.76671,+1	1.62640,3.92648,+1	0.25116,-2.55016,+1	-0.48612,0.72753,+1
-4.88488,-3.26397,+1	1.97397,3.69304,+1	0.00000,-2.50000,+1	-0.38300,0.71656,+1
-5.12616,-2.73999,+1	2.29172,3.42981,+1	-0.23891,-2.42576,+1	-0.28701,0.69290,+1
-5.31230,-2.20042,+1	2.57722,3.14035,+1	-0.46333,-2.32936,+1	-0.19957,0.65789,+1
-5.44259,-1.65099,+1	2.82842,2.82842,+1	-0.67128,-2.21292,+1	-0.12193,0.61299,+1
-5.51691,-1.09738,+1	3.04372,2.49792,+1	-0.86103,-2.07872,+1	-0.05513,0.55979,+1
-5.53571,-0.54522,+1	3.22194,2.15283,+1	-1.03118,-1.92920,+1	-0.00000,0.50000,+1
-5.50000,-0.00000,+1	3.36232,1.79720,+1	-1.18058,-1.76687,+1	0.00000,6.50000,-1
-5.41131,0.53296,+1	3.46454,1.43506,+1	-1.30843,-1.59433,+1	0.63098,6.40650,-1
-5.27172,1.04861,+1	3.52871,1.07042,+1	-1.41421,-1.41421,+1	1.24370,6.25250,-1
-5.08374,1.54213,+1	3.55534,0.70720,+1	-1.49770,-1.22913,+1	1.83242,6.04068,-1
-4.85036,2.00908,+1	3.54534,0.34918,+1	-1.55900,-1.04169,+1	2.39177,5.77424,-1
-4.57496,2.44537,+1	3.50000,0.00000,+1	-1.59848,-0.85440,+1	2.91676,5.45688,-1
-4.26128,2.84729,+1	3.42094,-0.33693,+1	-1.61678,-0.66969,+1	3.40286,5.09275,-1
-3.91336,3.21161,+1	3.31015,-0.65842,+1	-1.61483,-0.48985,+1	3.84600,4.68637,-1
-3.53553,3.53553,+1	3.16986,-0.96156,+1	-1.59377,-0.31702,+1	4.24264,4.24264,-1
-3.13231,3.81673,+1	3.00260,-1.24372,+1	-1.55497,-0.15315,+1	4.58974,3.76671,-1
-2.70840,4.05341,+1	2.81112,-1.50257,+1	-1.50000,-0.00000,+1	4.88488,3.26397,-1

5.12616,2.73999,-1	-0.43495,-4.41613,-1	-2.36734,1.94282,-1	1.61483,0.48985,-1
5.31230,2.20042,-1	-0.85352,-4.29093,-1	-2.12132,2.12132,-1	1.59377,0.31702,-1
5.44259,1.65099,-1	-1.25185,-4.12680,-1	-1.86353,2.27071,-1	1.55497,0.15315,-1
5.51691,1.09738,-1	-1.62640,-3.92648,-1	-1.59726,2.39047,-1	1.50000,0.00000,-1
5.53571,0.54522,-1	-1.97397,-3.69304,-1	-1.32580,2.48040,-1	1.43057,-0.14089,-1
5.50000,0.00000,-1	-2.29172,-3.42981,-1	-1.05237,2.54066,-1	1.34857,-0.26824,-1
5.41131,-0.53296,-1	-2.57722,-3.14035,-1	-0.78014,2.57177,-1	1.25598,-0.38099,-1
5.27172,-1.04861,-1	-2.82842,-2.82842,-1	-0.51211,2.57456,-1	1.15484,-0.47835,-1
5.08374,-1.54213,-1	-3.04372,-2.49792,-1	-0.25116,2.55016,-1	1.04728,-0.55978,-1
4.85036,-2.00908,-1	-3.22194,-2.15283,-1	-0.00000,2.50000,-1	0.93540,-0.62501,-1
4.57496,-2.44537,-1	-3.36232,-1.79720,-1	0.23891,2.42576,-1	0.82132,-0.67404,-1
4.26128,-2.84729,-1	-3.46454,-1.43506,-1	0.46333,2.32936,-1	0.70710,-0.70710,-1
3.91336,-3.21161,-1	-3.52871,-1.07042,-1	0.67128,2.21292,-1	0.59474,-0.72469,-1
3.53553,-3.53553,-1	-3.55534,-0.70720,-1	0.86103,2.07872,-1	0.48612,-0.72753,-1
3.13231,-3.81673,-1	-3.54534,-0.34918,-1	1.03118,1.92920,-1	0.38300,-0.71656,-1
2.70840,-4.05341,-1	-3.50000,-0.00000,-1	1.18058,1.76687,-1	0.28701,-0.69290,-1
2.26859,-4.24424,-1	-3.42094,0.33693,-1	1.30843,1.59433,-1	0.19957,-0.65789,-1
1.81774,-4.38842,-1	-3.31015,0.65842,-1	1.41421,1.41421,-1	0.12193,-0.61299,-1
1.36070,-4.48565,-1	-3.16986,0.96156,-1	1.49770,1.22913,-1	0.05513,-0.55979,-1
0.90229,-4.53613,-1	-3.00260,1.24372,-1	1.55900,1.04169,-1	0.00000,-0.50000,-1
0.44720,-4.54053,-1	-2.81112,1.50257,-1	1.59848,0.85440,-1	
0.00000,-4.50000,-1	-2.59834,1.73615,-1	1.61678,0.66969,-1	

BIBLIOGRAFIA

- Abe, S.; Thawonmas, R. (1997). "A fuzzy classifier with ellipsoidal regions". *IEEE Transactions on Fuzzy Systems* 5(3), pp. 358-368.
- Angelov, P. P. (2003). "An evolutionary approach to fuzzy rule-based model synthesis using indices for rules". *Fuzzy Sets and Systems* 137, pp. 325-338.
- Atmar, W. (1994). "Notes on the simulation of evolution". *IEEE Transactions on Neural Networks*, Vol. 5, N. 1, pp. 130-148.
- Bäck, T. (1996). "Evolutionary algorithms in theory and practice". *Oxford University Press*.
- Bäck, T. (2000). "Introduction to evolutionary algorithms". In *Evolutionary Computation 1*, Bäck, T.; Fogel, D.; Michalewicz, T. (eds), Institute of Physics Publishing.
- Baker, J. E. (1987). "Reducing bias and inefficiency in the selection algorithm". In *Proc. of the Second International Conference on Genetic Algorithm and Their Applications*, pp. 14-21, Hillsdale – New Jersey.
- Beasley, D. (2000). "Possible applications of evolutionary applications". In *Evolutionary Computation 1*, Bäck, T.; Fogel, D.; Michalewicz, T. (eds), Institute of Physics Publishing.
- Berenji, H. R.; Khedkar, P. (1992). "Learning and tuning fuzzy logic controller through reinforcement". *IEEE Trans. Neural Networks*, Vol. 3, N. 3, pp. 724-740.

- Berg, J.; Kaymak, U.; Bergh, W. (2002). “Fuzzy classification using probability-based rule weighting”. In *Proceedings of the 11th IEEE International Conference on Fuzzy Systems*, Vol. 2, pp. 991-996, Hawaii – USA.
- Bishop, C. M. (1995). “Neural networks for pattern recognition”. *Clarendon Press*.
- Bittencourt, G. (2006). “Inteligência artificial – Ferramentas e teorias”. *Editora UFSC*, 3a. edição.
- Bonarini, A. (1996). “Evolutionary learning of fuzzy rules: Competition and cooperation”. In W. Pedrycz, editor, *Fuzzy Modeling: Paradigms and Practice*, pp. 265-284, Kluwer Academic Press, Norwell, MA.
- Bonissone, P.; Khedkar, P. S.; Chen, Y. T. (1996). “Genetic algorithms for automated tuning of fuzzy controllers, a transportation application”. *Proceedings of the 5th International Conference on Fuzzy Systems (FUZZ-IEEE’96)*, pp. 674-680.
- Bucci, A. & Pollack, J. (2003). “A Mathematical Framework for the Study of Coevolution”. In C. Cotta, K. De Jong, R. Poli, & J. Rowe (Eds.), *Foundations of Genetic Algorithms 7 (FOGA VII)*. Morgan Kaufmann, pp. 221-235.
- Bucci, A.; Pollack, J.; De Jong, E. (2004). “Automated Extraction of Problem Structure”. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2004)*, Springer-Verlag, June 26-30, Seattle, USA, pp. 501-512.
- Bucci, A. & Pollack, J. (2005). “On Identifying Global Optima in Cooperative Coevolution”. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2005)*, ACM Press, June 25-29, Washington, DC, USA, pp. 539-544.
- Carse, B.; Forgarty, T. C.; Munro, A. (1996). “An evolving fuzzy rule based controllers using genetic algorithms”. *Fuzzy Sets and Fuzzy Systems*, Vol. 80, pp. 273-293.
- Carvalho, A.; Braga, A.; Ludermir, T. (2003). “Computação evolutiva”. In *Sistemas Inteligentes – Fundamentos e Aplicações*, Resende, S. (eds), Ed. Manole.
- Casillas, J.; Cordón, O.; Herrera, F. (2000). “A methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules”. *Technical Report #DECSAI-000 101*, University of Granada – Spain.

- Casillas, J.; Cordón, O.; Jesus, M.; Herrera, F. (2005). “Genetic Tuning of Fuzzy Rule Deep Structures Preserving Interpretability and Its Interaction With Fuzzy Rule Set Reduction”. *IEEE Transactions on Fuzzy Systems*, Vol. 13, N. 1, pp. 13-29.
- Castellano, G.; Fanelli, A. M. (2000). “A staged approach for generation and compression of fuzzy classification rules”. *In Proceedings of FUZZ-IEEE’2000*, pp. 42-47, San Antonio, USA.
- Castro, P. A.; Pires, M. G.; Camargo, H. C. (2003). “Aprendizado e seleção de regras fuzzy usando algoritmos genéticos”. *VI SBAI – Simpósio Brasileiro de Automação Inteligente*, Bauru, SP.
- Castro, P. A.; Camargo, H. C. (2004). “A study of the reasoning methods impact on genetic learning and optimization of fuzzy rules”. *SBIA 2004 – Brazilian Symposium on Artificial Intelligence*, Vol. 3171, pp. 414-423, São Luis, MA.
- Castro, P. A.; Pires, M. G.; Camargo, H. C. (2004a). “Genetic design of fuzzy knowledge bases – a study of different approaches”. *NAFIPS 2004 – North American Fuzzy Information Processing Society*, Vol. 2, pp. 954-959, Banff, Canada.
- Castro, P. A.; Pires, M. G.; Camargo, H. C. (2004b). “Learning and optimization of fuzzy rule base by means of self-adaptive genetic algorithm”. *FUZZ-IEEE 2004 – IEEE International Conference on Fuzzy Systems*, Vol. 2, pp. 1037-1042, Budapest, Hungria.
- Castro, P. A.; Pires, M. G.; Camargo, H. C.; Morandin, O.; Kato, E. R. (2004c). “Genetic learning of fuzzy rules applied to sequencing problem of FMS”. *IEEE SMC – IEEE International Conference on Systems, Man and Cybernetics*, Vol. 5, pp. 4336-4341, The Hague, The Netherlands.
- Castro, P. A.; Santoro, D. M.; Camargo, H. C.; Nicoletti, M. C. (2004d). “Improving a Pittsburgh learnt fuzzy rule base using feature subset selection”. *HIS 2004 – Fourth International Conference on Hybrid Intelligent Systems*, pp. 180-185, Kitakyushu, Japan.
- Castro, P. A.; Camargo, H. C. (2005a). “Focusing on interpretability and accuracy of a genetic fuzzy system”. *Fuzz-IEEE 2005 – IEEE International Conference on Fuzzy Systems*, Reno-Nevada, 22-25 May 2005.

- Castro, P. A.; Camargo, H. C. (2005b). "Improving the genetic optimization of fuzzy rule base by imposing a constraint condition on the number of rules". *Encontro Nacional de Inteligência Artificial (ENIA 2005)*, pp. 972-981, São Leopoldo, RS.
- Chang, X.; Lilly, J. H. (2004). "Evolutionary design of a fuzzy classifier from data". *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, Vol. 34, N. 4, pp. 1894-1906.
- Chen, S.; Huang, C. (2003). "Generation weighted fuzzy rules from relational database systems for estimating null values using genetic algorithms". *IEEE Transactions on Fuzzy Systems*, Vol. 11, N. 4, pp. 495-506.
- Chie, Z.; Yan, H.; Pham, T. (1996). "Fuzzy algorithms with applications to image processing and pattern recognition". *World Scientific*.
- Chiou, Y.; Lan, L. W. (2005). "Genetic fuzzy logic controller: an iterative evolution algorithm with new encoding method". *Fuzzy Sets and Systems 152*, pp. 617-635.
- Codd, E. F. (1970). "A relational model of data for large shared data banks". *Communications of the ACM*, Vol. 13, N. 6, pp. 377-387.
- Cordón, O.; Herrera, F. (1997). "A three-stage evolutionary process for learning descriptive and approximate fuzzy logic controller knowledge bases from examples". *International Journal of Approximate Reasoning*, Vol. 7, N. 4, pp. 369-407.
- Cordón, O.; Herrera, F. (2000). "A proposal for improving the accuracy of linguistic modeling". *IEEE Transactions on Fuzzy Systems*, Vol. 8, pp. 335-344.
- Cordón, O.; Herrera, F. (2001). "Hybridizing genetic algorithms with sharing scheme and evolution strategies for designing approximate fuzzy rule-based systems". *Fuzzy Sets and Systems 118*, pp. 235-255.
- Cordón, O.; Herrera, F.; Hoffmann, F.; Magdalena, L. (2001a). "Genetic fuzzy systems – evolutionary tuning and learning of fuzzy knowledge bases". *World Scientific*.
- Cordón, O.; Herrera, F.; Magdalena, L.; Villar, P. (2001b). "A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base". *Journal of Information Sciences 136*, pp. 85-107.

- Cordón, O.; Herrera, F.; Villar, P. (2001c). “Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base”. *IEEE Transactions on Fuzzy Systems* 9, pp. 667-674.
- Cordón, O.; Gomide, F.; Herrera, F.; Hoffmann, F.; Magdalena, L. (2004a). “Genetic fuzzy systems: New developments”. *Fuzzy Sets and Systems*, Vol. 141, pp. 1-3.
- Cordón, O.; Herrera, F.; Gomide, F.; Hoffmann, F.; Magdalena, L. (2004b). “Ten years of genetic fuzzy systems: Current framework and new trends”. *Fuzzy Sets and Systems*, Vol. 141, pp. 5-31.
- Cruz, A. V. (2003). “Otimização de planejamentos com restrição de precedência usando algoritmos genéticos e co-evolução cooperativa”. *PhD thesis*, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro - Brasil.
- Darwin, C. (1859). “On the origin of species by means of natural selection”. *London*.
- De Jong, K. A. (1975). “The analysis and behavior of a class of genetic adaptive system”. *PhD thesis*, Department of Computer and Communications, University of Michigan.
- De Jong, E. & Pollack, J. (2003). “Learning the Ideal Evaluation Function”. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO2003), Springer-Verlag, July 12-16, Chicago, USA, pp. 274-285.
- De Jong, E. & Bucci, A. (2006). “DECA: Dimension Extracting Coevolutionary Algorithm”. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO2006), ACM Press, July 8-12, Seattle, USA, pp. 313-320.
- Delgado, M. R.; Zuben, F. V.; Gomide, F. (2001). “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. (2002). “Projeto automático de sistemas nebulosos: Uma abordagem co-evolutiva”. *PhD thesis*, Universidade Estadual de Campinas, Campinas - Brasil.
- Delgado, M. R.; Zuben, F. V.; Gomide, F. (2002). “Coevolutionary design of takagi-sugeno fuzzy systems”. *Proceedings of 2002 World Congress on Computational Intelligence*, Vol. 2, pp. 1384-1389.

- Delgado, M. R.; Zuben, F. V.; Gomide, F. (2004). "Coevolutionary genetic fuzzy systems: A hierarchical collaborative approach". *Fuzzy Sets and Systems 141*, pp. 89-106.
- Duda, R. O.; Hart, P. E. (1973). "Pattern classification and scene analysis". *John Wiley*.
- Dumitrescu, D.; Lazzerini, B.; Jain, L. L. (2000). "Fuzzy sets and their application to clustering and training". *International Series on Computational Intelligence*, CRC Press, Boca Raton.
- Ehrlich, P. R.; Raven, P. H. (1964). "Butterflies and plants: A study in coevolution". *Evolution*, Vol. 18, pp. 586-608.
- Ficici, S.; Pollack, J. (2003). "A Game-Theoretic Memory Mechanism for Coevolution". Proceedings of the Evolutionary Computation Conference (GECCO 2003), Springer-Verlag, July 12-16, Chicago, USA, pp. 286-297.
- Ficici, S.; Melnik, O.; Pollack, J. (2005). "A Game-Theoretic and Dynamical Analysis of Selection Methods in Coevolution". *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 6, pp. 580-602.
- Filipic, B.; Juricic, D. (1996). "A genetic algorithm to support learning fuzzy control rules from examples". *Genetic Algorithms and Soft Computing*, pp. 403-418, F. Herrera and J. L. Verdegay, Eds: Physica-Verlag.
- Fogel, D. B. (1995). "Evolutionary computation – Toward a new philosophy of machine intelligence". *IEEE Press*.
- Fogel, D. B.; Owens, A. J.; Walsh, M. J. (1996). "Artificial intelligent through simulated evolution". *John Wiley*.
- Forrest, S.; Perelson, A. S. (1990). "Genetic algorithms and the immune system". *Parallel Problem Solving from Nature*, In H. P. Schwefel and R. Männer (eds), pp. 320-325, Springer-Verlag.
- Forrest, S.; Javornik, B.; Smith, R. E.; Perelson, A. S. (1993). "Using genetic algorithms to explore pattern recognition in the immune system". *Evolutionary Computation 1(3)*, pp. 191-211.

- Garis, H. (1990). "Building artificial nervous systems using genetically programmed neural network modules". In Porter & Mooney (Eds.), *Proceeding of the Seventh International Conference on Machine Learning*, pp. 132-139, Morgan Kaufmann.
- Geffeney, S. L.; Fujimoto, E.; Brodie, E. D.; Brodie Jr, E. D.; Ruben, P. C. (2005). "Evolutionary diversification of TTX-resistant sodium channels in a predator-prey interaction". *Natura* 434, pp. 759-763.
- Giordana, A.; Saitta, L; Zini, F. (1994). "Learning disjunctive concepts by means of genetic algorithms". In Cohen & Hirsh (Eds.), *Proceedings of the Eleventh International Conference on Machine Learning*, pp. 96-104, Morgan Kaufmann.
- Giordana, A.; Neri, F. (1996). "Search-intensive concept induction". *Evolutionary Computation* 3(4), pp. 375-416.
- Goldberg, D. E. (1989). "Genetic algorithms in search, optimization, and machine learning". Addison-Wesley, Reading, MA.
- Gonzalez, A.; Perez, R. (1999). "SLAVE: A genetic learning system based on an iterative approach". *IEEE Transaction on Fuzzy Systems* 7(2), pp. 176-191.
- Gudwin, R.; Gomide, F.; Pedrycz, W. (1998). "Context adaptation in fuzzy processing and genetic algorithms". *International Journal of Intelligent Systems* 13, pp. 929-948.
- Gurocak, H. B. (1999). "A genetic-algorithm-based method for tuning fuzzy logic controllers". *Fuzzy Sets and Systems* 108 (1), pp. 39-47.
- Herrera, F.; Lozano, M.; Verdegay, J. L. (1995). "Tuning fuzzy logic controllers by genetic algorithms". *International Journal of Approximate Reasoning* 12, pp. 299-315.
- Hillis, W. D. (1991). "Co-Evolving parasites improve simulated evolution as an optimization procedure". *Artificial Life II, SFI Studies in the Sciences of Complexity*, In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen (eds), Vol. 10, pp. 313-324.
- Hoffmann, F.; Pfister, G. (1997). "Evolutionary design of a fuzzy knowledge base for a mobile robot". *International Journal of Approximate Reasoning*, Vol. 17, pp. 447-469.

- Hoffmann, F. (2004). "Combining boosting and evolutionary algorithms for learning of fuzzy classification rules". *Fuzzy Sets and Systems 141*, pp. 47-58.
- Holland, J. H. (1975). "Adaptation in natural and artificial systems". *University of Michigan Press*.
- Holland, J. H.; Reitman, J. S. (1978). "Cognitive systems based on adaptive algorithms". *Pattern-Directed Inference Systems*, In D. A. Waterman and F. Hayes-Roth (eds), Academic Press.
- Holland, J. H. (1986). "Escaping brittleness: The possibilities of general purpose learning algorithms applied to parallel rule-based systems". *Machine Learning*, In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (eds), Vol. 2, pp. 593-623.
- Homaifar, A.; McCormick, E. (1995). "Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms". *IEEE Transactions on Fuzzy Systems 3*, pp. 129-139.
- Hwang, W. R.; Thompson, W. E. (1994). "Design of fuzzy logic controllers using genetic algorithms". *Proceedings of the 3rd International Conference Fuzzy Systems (FUZZ-IEEE'94)*, pp. 1383-1388, Orland – FL.
- Ishibuchi, H.; Nozaki, K.; Tanaka, H.; Hosaka, Y.; Matsuda, M. (1994). "Empirical study on learning in fuzzy systems by rice analysis". *Fuzzy Sets and Systems*, Vol. 64, pp. 129-144.
- Ishibuchi, H.; Murata, T. (1996). "A genetic-algorithm-based fuzzy partition method for pattern classification problems". *Genetic Algorithms and Soft Computing*, pp. 555-578, Berlin, Germany, Physica-Verlag.
- Ishibuchi, H.; Murata, T.; Turksen, I. B. (1997). "Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classifications problems". *Fuzzy Sets and Systems 89*, pp. 134-150.
- Ishibuchi, H.; Nakashima, T.; Murata, T. (1999a). "Voting fuzzy rule-based systems for pattern classification problems". *Fuzzy Sets and Systems 103*, pp. 223-238.
- Ishibuchi, H.; Nakashima, T.; Murata, T. (1999b). "Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems". *IEEE Transactions on Fuzzy Systems 29*, pp. 601-618.

- Ishibuchi, H.; Yamamoto, T. (2004). "Fuzzy rule by multi-objective genetic local search algorithms and rule evaluation measures in data mining". *Fuzzy Sets and Systems* 141, pp. 59-88.
- 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*.
- Jesus, M. J.; Hoffmann, F.; Navascues, L. J.; Sanchez, L. (2004). "Induction of fuzzy-rule-based classifiers with evolutionary boosting algorithms". *IEEE Transactions on Fuzzy Systems*, Vol. 12, N. 3, pp. 296-308.
- 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.
- Juang, C. (2005). "Combination of online clustering and Q-Value based GA for reinforcement fuzzy system design". *IEEE Transactions on Fuzzy Systems*, Vol. 13, N. 3, pp. 289-302.
- Karr, C. L. (1991a). "Design of an adaptive fuzzy logic controller using a genetic algorithm". *Proceedings of the 4th International Conference Genetic Algorithms*, pp. 450-457, University of California, San Diego, July 13-16.
- Karr, C. L. (1991b). "Genetic algorithm for fuzzy controllers". *AI Expert*, pp. 26-33.
- Kecman, V. (2001). "Learning and soft computing". *MIT Press*.
- Kim, M.; Kim, C.; Lee, J. (2006). "Evolving compact and interpretable Takagi-Sugeno fuzzy models with a new encoding scheme". *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, Vol. 36, N. 5, pp. 1006-1023.
- Klement, E. P.; Mesiar, R.; Pap, E. (2000). "Triangular norms". *Kluwer Academic Publishers*, c2000.
- Klir, G.; Yuan, B. (1995). "Fuzzy sets and fuzzy logic – Theory and applications". *Prentice-Hall*.
- Koza, J. R. (1992). "Genetic programming: On the programming of computers by means of natural selection". *MIT Press*.

- Koza, J. R. (1993). "Hierarchical automatic function definition in genetic programming". *Foundations of Genetic Algorithms 2*, In L. D. Whitley (ed), pp. 297-318, Morgan Kaufmann.
- 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. (1993a). "Embedding apriori knowledge into an integrated fuzzy system design method based on genetic algorithm". *Proceeding of the 5th IFSA World Congress (IFSA '93)*, pp. 1293-1296, Seoul, Korea.
- Lee, M. A.; Takagi, H. (1993b). "Integrating design stages of fuzzy systems using genetic algorithms". *Proceedings of FUZZ-IEEE'93*, pp. 612-617, San Francisco, USA.
- Lee, C. W.; Shin, Y. C. (2003). "Construction of fuzzy systems using least-squares method and genetic algorithm". *Fuzzy Sets and Systems 137*, pp. 297-323.
- Liao, T. W.; Celmins, A. K.; Hammenll, R. J. (1997). "A fuzzy c-means variant for the generation of fuzzy terms sets". *Fuzzy Sets and Systems 135*, pp. 241-257.
- Luger, G. (2002). "Artificial intelligence: Structures and strategies for complex problem solving". *Addison Wesley Longman, Inc.*
- Lukasiewicz, J. (1970). "Philosophical remarks on many-valued systems of propositional logic". *In Studies in Logic and the Foundations of Mathematics*, North Holland.
- Magdalena, L.; Velasco, J. R. (1996). "Fuzzy rules-based controllers that learn by evolving its knowledge base". *In Herrera, F.; Verdegary, J. L. (eds), Fuzzy Logic and Soft Computing*, Physica-Verlag.
- Magdalena, L.; Monasterio, F. (1997). "A fuzzy logic controller with learning through the evolution of its knowledge base". *International Journal Approximate Reasoning 16(3-4)*, pp. 335-358.
- Magdalena, L.; Velasco, J. R. (1997). "Evolutionary based learning of fuzzy controllers". *In Pedrycz, W. (ed), Fuzzy Evolutionary Computation*, Kluwer Academic Publisher.

- 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.
- Mamdani, E. H. (1977). "Application of fuzzy logic to approximate reasoning using linguist systems". *Fuzzy Sets and Systems* 26, pp. 1182-1191.
- Maruo, M. H.; Delgado, M. R. (2006). "Co-evolutionary genetic fuzzy system: a self-adapting approach". *IEEE International Conference on Fuzzy Systems*, Vancouver, BC, Canada.
- Merz, C. J.; Murphy, P. M. (1998). "UCI repository of machine learning database". <http://www.ics.uci.edu/~mlearn>, University of California, Department of Information and Computer Science, Irvine, CA.
- Michalewicz, Z. (1996). "Genetic algorithms + data structures = Evolution programs". *Springer-Verlag*.
- Mitchell, M. (1996). "An introduction to genetic algorithms". *MIT Press*.
- Moriarty, D. E. (1997). "Symbiotic evolution of neural networks in sequential decision tasks". *PhD thesis*, University of Texas, Austing, Texas.
- Moriarty, D. E.; Miikkulainen, R. (1997). "Forming neural networks through efficient and adaptive convolution". *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.
- Nauck, D.; Kruse, R. (1997). "A neuro-fuzzy method to learn fuzzy classification rules from data". *Fuzzy Sets and Systems* 89, pp. 277-288.
- Nomura, H.; Hayashi, L.; Wakami, N. (1992). "A learning method of fuzzy inference rules by descent method". *In Proceedings of the 1st IEEE International Conference on Fuzzy Systems*, pp. 203-210, San Diego – USA.
- Olsson, B. (2001). "Co-evolutionary search in asymmetric space". *Information Sciences*, Vol. 133, pp. 103-125.

- Parodi, A.; Bonelli, P. (1993). "A new approach to fuzzy classifier systems". In *Proceedings of Fifth International Conference on Genetic Algorithms (ICGA'93)*, Morgan Kaufmann, Los Altos, CA, pp. 223-230.
- Pedrycz, W. (1996). "Fuzzy modelling: Paradigms and Practice". *Kluwer Academic Press*.
- Pedrycz, W. (1997). "Fuzzy evolutionary computation". *Kluwer Academic Publishers*, Dordrecht.
- Pedrycz, W. (1998). "Computation intelligence – An introduction". *CRC Press*, Boca Raton.
- Pedrycz, W.; Gomide, F. (1998). "An introduction to fuzzy sets: Analysis and design". *MIT Press*, Cambridge.
- Pena-Reyes, C. A.; Sipper, M. (2001). "Fuzzy CoCo: A cooperative-coevolutionary approach of fuzzy modeling". *IEEE Transactions on Fuzzy Systems*, Vol. 9, N. 5, pp. 727-737.
- Pires, M. G.; Camargo, H. C. (2004). "Genetic learning and optimization of fuzzy sets in fuzzy rule-based systems". *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration (IRI-2004)*, Las Vegas, Nevada, USA, pp. 623-628.
- Potter, M. A.; De Jong, K. A. (2000). "Cooperative coevolution: An architecture for evolving coadapted subcomponents". *Evolutionary Computation* 8(1), pp. 1-29, The MIT Press.
- Quinlan, J. R. (1986). "Induction of decision trees". *Machine Learning*, Vol. 1, pp. 81-106.
- Rechenberg, I. (1965). "Cybernetic solution path of an experimental problem". *Technical report*, Royal Aircraft Establ, Farnborough, UK.
- Resende, S. (2003). "Sistemas inteligentes – Fundamentos e aplicações". *Editores Manole*.
- Rivas, V. M.; Merelo, J. J.; Rojas, I.; Romero, G.; Castillo, P. A.; Carpio, J. (2003). "Evolving two-dimensional fuzzy systems". *Fuzzy Sets and Systems* 138, pp. 381-398.

- Rosca, J. P.; Ballard, D. H. (1994). "Hierarchical self-organization in genetic programming". *Proceedings of the Eleventh International Conference on Machine Learning*, In W. Cohen and H. Hirsh (eds), pp. 251-258, Morgan Kaufmann.
- Rosca, J. P.; Ballard, D. H. (1996). "Discovery of subroutines in genetic programming". *Advances in Genetic Programming 2*, In P. Angeline and K. E. Kinnear (eds), Chapter 9, The MIT Press.
- Rosin, C.; Belew, R. W. (1997). "New methods for competitive coevolution". *Evolutionary Computation*, Vol. 5, N. 1, pp. 1-29.
- Sanchez, E.; Shibata, T.; Zadeh, L. (1997). "Genetic algorithms and fuzzy logic systems". *Soft Computing Perspective*, World Scientific, Singapore.
- Setnes, M.; Roubos, H. (2000). "GA-fuzzy modeling and classification: complexity and performance". *IEEE Trans. Fuzzy Systems* 8 (5), pp. 509-522.
- Sim, K.; Byun, K.; Lee, D. (2004). "Design of Fuzzy Controller Using Schema Coevolutionary Algorithm". *IEEE Transactions on Fuzzy Systems*, Vol. 12, N. 4, pp. 565-568.
- Smith, R. E.; Forrest, S.; Perelson, A. S. (1993). "Searching for diverse, cooperative populations with genetic algorithms". *Evolutionary Computation* 1(2), pp. 127-149.
- Spiegel, D.; Sudkamp, T. (2003). "Sparse data in the evolutionary generation of fuzzy models". *Fuzzy Sets and Systems* 138, pp. 363-379.
- 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.
- Takagi, T.; Sugeno, M. (1985). "Fuzzy identification of systems and its application to modeling and control". *IEEE Trans. on Systems, Man, and Cybernetics* 15(1), pp. 116-132.
- Teng, Y.; Wang, W. (2004). "Constructing a user-friendly GA-Based fuzzy system directly from numerical data". *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, Vol. 34, N. 5, pp. 2060-2070.
- Velasco, J. R. (1998). "Genetic-based on-line learning for fuzzy process control". *International Journal Intelligent System* 13(10-11), pp. 891-903.

- Wang, L. X.; Mendel, J. M. (1992). "Generation fuzzy rules by learning from examples". *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 22, pp. 1414-1427.
- Wang, H.; Kwong, S.; Jin, Y.; Wei, W.; Man, K. F. (2005). "Multi-objective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction". *Fuzzy Sets and Systems 149*, pp. 149-186.
- Yager, R.; Filev, D. (1994). "Essentials of fuzzy modeling and control". *John Wiley & Sons*.
- Yuan, Y.; Zhuang, H. (1996). "A genetic algorithm for generating fuzzy classification rules". *Fuzzy Sets and Systems 84(4)*, pp. 1-19.
- Zadeh, L. A. (1965). "Fuzzy sets". *Information and Control 8(3)*, pp. 338-353.
- Zadeh, L. A. (1973). "Outline of a new approach to the analysis of complex system and decision process". *IEEE Trans. on Systems, Man, and Cybernetics 3*, pp. 28-44.
- Zimmermann, H. J. (1991). "Fuzzy set theory and its applications". *Kluwer Academic Publishers*.
- Zong-Yi, X.; Yuan-Long, H.; Yong, Z.; Li-Min, J.; Yuexiam, H. (2006). "A multi-objective cooperative coevolutionary algorithm for constructing accurate and interpretable fuzzy systems". *IEEE International Conference on Fuzzy Systems*, Vancouver, BC, Canada.