

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**GERAÇÃO GENÉTICA MULTIOBJETIVO DE
BASES DE CONHECIMENTO *FUZZY* COM
ENFOQUE NA DISTRIBUIÇÃO DAS
SOLUÇÕES NÃO DOMINADAS**

ADINOVAM HENRIQUES DE MACEDO PIMENTA

ORIENTADORA: PROFA. DRA. HELOISA DE ARRUDA CAMARGO

São Carlos – SP

Novembro/2014

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**GERAÇÃO GENÉTICA MULTIOBJETIVO DE
BASES DE CONHECIMENTO *FUZZY* COM
ENFOQUE NA DISTRIBUIÇÃO DAS
SOLUÇÕES NÃO DOMINADAS**

ADINOVAM HENRIQUES DE MACEDO PIMENTA

Tese apresentada ao Programa de Pós-Graduação em
Ciência da Computação da Universidade Federal de
São Carlos, como parte dos requisitos para a obten-
ção do título de Doutor em Ciência da Computação,
área de concentração: Inteligência Artificial
Orientadora: Profa. Dra. Heloisa de Arruda Ca-
margo

São Carlos – SP

Novembro/2014

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária UFSCar
Processamento Técnico
com os dados fornecidos pelo(a) autor(a)

P644g Pimenta, Adinovam Henriques de Macedo
Geração genética multiobjetivo de bases de conhecimento Fuzzy com enfoque na distribuição das soluções não dominadas / Adinovam Henriques de Macedo Pimenta. -- São Carlos : UFSCar, 2017.
110 p.

Tese (Doutorado) -- Universidade Federal de São Carlos, 2014.

1. Algoritmos genéticos. 2. Algoritmos genéticos multiobjetivo. 3. Sistemas Fuzzy genéticos. 4. Geração automática de regras Fuzzy. 5. Fronteira de Pareto. I. Título.



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Tese de Doutorado do candidato Adinovam Henriques de Macedo Pimenta, realizada em 02/12/2014:

Profa. Dra. Heloisa de Arruda Camargo
UFSCar

Prof. Dr. Marley Maria Bernardes Rebuszi Vellasco
PUC/RJ

Prof. Dr. Ricardo Tanscheit
PUC/RJ

Profa. Dra. Solange Oliveira Rezende
USP

Profa. Dra. Maria do Carmo Nicoletti
UFSCar

À minha mãe Alaide Henriques de Macedo Pimenta (*in memoriam*).

AGRADECIMENTOS

A Deus, por ter me dado este sonho e os recursos necessários para torná-lo em realidade.

À minha amada esposa Mayra, pelo amor, carinho, apoio, parceria e compreensão nas fases mais difíceis deste trabalho. Obrigado por me mostrar que nem sempre devemos optar pelo caminho mais fácil. Você é o meu maior incentivo.

Ao meu pai Ernane, à minha mãe Alaide, à minha irmã Djanane e à minha avó Francisca, pelo apoio, incentivo e cuidado que sempre me deram.

À Prof.^a Dr.^a Heloisa de Arruda Camargo por sua orientação, pela disposição e paciência.

Aos meus amigos Priscilla de Abreu Lopes e José Santana pela grande ajuda que me deram na reta final deste trabalho.

Aos meus irmãos de laboratório Cristiane Yahuinuma, Edward Cardenas, Fábio Justo, Helano Povoas, Marcos Cintra e Mariana Ribeiro. Todo tipo de ajuda que eu tive de vocês foi muito importante para mim. Muito obrigado!

A todos que contribuíram de forma direta ou indireta para a realização deste trabalho.

Will we ever know?

Ylvis

RESUMO

A construção da base de conhecimento de sistemas *fuzzy* tem sido beneficiada intensamente por métodos automáticos que extraem o conhecimento necessário a partir de conjuntos de dados que representam exemplos do problema. Entre os tópicos mais investigados nas pesquisas recentes está a questão do balanceamento entre acuidade e interpretabilidade, que têm sido abordada por meio dos algoritmos genéticos multiobjetivo, sendo o NSGA-II um dos mais populares. Neste escopo, identificou-se a necessidade do controle da distribuição das soluções encontradas por estes algoritmos, a fim de que cada solução possa equilibrar a fronteira de Pareto com relação aos objetivos otimizados pelo algoritmo genético multiobjetivo. Neste sentido, desenvolveu-se neste projeto de doutorado um algoritmo genético multiobjetivo, chamado NSGA-DO, capaz de encontrar soluções não dominadas que equilibram a fronteira de Pareto nos objetivos a serem otimizados. A principal característica do NSGA-DO é a seleção de soluções orientada à distância. Uma vez encontrada a fronteira de Pareto, o algoritmo usa a localização das soluções nesta fronteira para encontrar a melhor distribuição das soluções. Para a validação da proposta, aplicou-se o NSGA-DO em uma metodologia para a geração de bases de conhecimento *fuzzy*. Experimentos realizados comprovaram a superioridade do NSGA-DO com relação ao NSGA-II nos três quesitos analisados: dispersão, acurácia e interpretabilidade.

Palavras-chave: Algoritmos Genéticos, Algoritmos Genéticos Multiobjetivo, Sistemas *Fuzzy* Genéticos, Geração Automática de Regras *Fuzzy*, Fronteira de Pareto

ABSTRACT

The process of building the knowledge base of fuzzy systems has benefited extensively of methods to automatically extract the necessary knowledge from data sets that represent examples of the problem. Among the topics investigated in the most recent research is the matter of balance between accuracy and interpretability, which has been addressed by means of multi-objective genetic algorithms, NSGA-II being one of the most popular. In this scope, we identified the need to control the diversity of solutions found by these algorithms, so that each solution would balance the Pareto frontier with respect to the goals optimized by the multi-objective genetic algorithm. In this PhD thesis a multi-objective genetic algorithm, named NSGA-DO, is proposed. It is able to find non dominated solutions that balance the Pareto frontier with respect optimization of the objectives. The main characteristic of NSGA-DO is the distance oriented selection of solutions. Once the Pareto frontier is found, the algorithm uses the locations of the solutions in the frontier to find the best distribution of solutions. As for the validation of the proposal, NSGA-DO was applied to a methodology for the generation of fuzzy knowledge bases. Experiments show the superiority of NSGA-DO when compared to NSGA-II in all three issues analyzed: dispersion, accuracy and interpretability.

Keywords: Genetic Algorithms, Multiobjective Genetic Algorithms, Genetic *Fuzzy* Systems, Automatic Generation of *Fuzzy* Rule, Pareto-Optimal Front

LISTA DE FIGURAS

2.1	Caracterização da interpretabilidade. Figura baseada em (ALONSO; MAGDALENA; GONZÁLEZ-RODRÍGUEZ, 2009).	34
2.2	Partição <i>fuzzy</i> forte	35
2.3	Particionamento com baixo significado semântico	38
3.1	Método de seleção por roleta.	44
3.2	Cruzamento de 1 ponto.	45
3.3	Cruzamento de 2 pontos.	45
3.4	Exemplo de compra de carro, considerando o custo e o conforto.	48
3.5	Distribuição das soluções ao longo da fronteira de Pareto	49
3.6	Soluções Pareto-ótimo locais e globais.	51
3.7	Cálculo da distância de multidão do NSGA-II em um espaço bidimensional (DEB et al., 2002a)	55
3.8	Classificação dos Sistemas <i>Fuzzy</i> Genéticos. Adaptado de (HERRERA, 2008).	57
3.9	Aprendizagem genética simultânea dos componentes da BC. Adaptado de (HERRERA, 2008).	60
3.10	Classificação dos SFGMO. Adaptado de (FAZZOLARI et al., 2013).	61
4.1	Etapa 1 - Seleção de regras	65
4.2	Etapa 2 - Otimização da BD	66
4.3	Fronteira com uma única inflexão	69
4.4	Fronteira com muitas inflexões	70
4.5	Exemplos do uso do Índice de Distribuição	71

4.6	Diferença entre a escolha por PM e por PEF	74
4.7	Situação em que o algoritmo de corte do NSGA-II erra	75
4.8	Escolha de solução por proximidade ao ponto ideal	76
4.9	Representação cromossômica de uma BR	78
4.10	Representação cromossômica de uma BD	79
5.1	Problema de dominância	86
5.2	Dominância entre o NSGA-II e o NSGA-DO para a seleção de regras - Parte I.	87
5.3	Dominância entre o NSGA-II e o NSGA-DO para a seleção de regras - Parte II.	88
5.4	Dominância entre o NSGA-II e o NSGA-DO para a otimização da BD - Parte I.	89
5.5	Dominância entre o NSGA-II e o NSGA-DO para a otimização da BD - Parte II.	90

LISTA DE TABELAS

2.1	Taxonomia para a interpretabilidade. Traduzida de (GACTO; ALCALÁ; HERRERA, 2011)	39
3.1	Diferentes tipos de AGMO	53
5.1	Conjuntos do repositório <i>UCI</i> utilizados nos experimentos	83
5.2	Configuração dos parâmetros dos AGMOs	83
5.3	Dispersão das soluções na fronteira final	84
5.4	Taxa média das soluções centrais para a BR	85
5.5	Taxa média das soluções centrais para a BD	85
5.6	Número de soluções na fronteira ótimo de Pareto	91
5.7	Dominância do NSGA-II sobre o NSGA-DO na etapa de seleção de regras . . .	92
5.8	Dominância do NSGA-DO sobre o NSGA-II na etapa de seleção de regras . . .	93
5.9	Maior dominância por conjunto de dados na etapa 1	94
5.10	Dominância do NSGA-II sobre o NSGA-DO na etapa de otimização da BD . .	95
5.11	Dominância do NSGA-DO sobre o NSGA-II na etapa 2 (Otimização BD) . . .	95
5.12	Maior dominância por conjunto de dados na etapa 2	95
5.13	Teste Wilcoxon	96
6.1	Métodos propostos neste trabalho	99

ACRÔNIMOS E SIGLAS

AGMO – *Algoritmo Genético Multiobjetivo*

AG – *Algoritmo Genético*

BC – *Base de Conhecimento*

BD – *Base de Dados*

BR – *Base de Regras*

CE – *Computação Evolutiva*

EE – *Estratégia Evolutiva*

IC – *Inteligência Computacional*

ID – *Índice de Distribuição*

MI – *Mecanismo de Inferência*

NSGA-DO – *Non-dominated Sorting Genetic Algorithm Distance-Oriented*

NSGA-II – *Non-dominated Sorting Genetic Algorithm II*

PEF – *Ponto de Equilíbrio da Fronteira*

PG – *Programação Genética*

PM – *Ponto Médio*

SCF – *Sistema de Classificação Fuzzy*

SFBR – *Sistema Fuzzy Baseado em Regra*

SFG – *Sistema Fuzzy Genéticos*

SF – *Sistema Fuzzy*

iBR – *Índice de Interpretabilidade da Base de Regras*

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO	14
1.1 Motivações, hipótese e objetivo	15
1.2 Contribuições	16
1.3 Organização da Tese	17
CAPÍTULO 2 – SISTEMAS FUZZY	18
2.1 Considerações Iniciais	18
2.2 Conjuntos <i>Fuzzy</i>	18
2.2.1 Funções de Pertinência	19
2.2.2 Operações Básicas sobre Conjuntos <i>Fuzzy</i>	21
2.2.3 Relações <i>Fuzzy</i>	24
2.2.4 Composição de Relações <i>Fuzzy</i>	25
2.3 Regras <i>Fuzzy</i>	26
2.3.1 Variáveis Linguísticas	27
2.3.2 Sintaxe e Semântica das Regras <i>Fuzzy</i>	28
2.4 Sistemas de Inferência <i>Fuzzy</i>	29
2.5 Sistema de Classificação <i>Fuzzy</i>	31
2.6 Interpretabilidade de Sistemas <i>Fuzzy</i> Baseados em Regras	32
2.6.1 Taxonomia para Índices de Interpretabilidade de SFBR	34
2.7 Considerações finais	39

CAPÍTULO 3 – COMPUTAÇÃO EVOLUTIVA	41
3.1 Considerações Iniciais	41
3.2 Algoritmos Genéticos	42
3.2.1 Avaliação	43
3.2.2 Seleção	43
3.2.3 Cruzamento	44
3.2.4 Cruzamento Aritmético	45
3.2.5 Mutação	45
3.3 Otimização Multiobjetivo	46
3.3.1 Soluções Pareto-ótimas	47
3.3.2 Metas em Problemas de Otimização Multiobjetivo	48
3.3.3 Diferenças entre Otimização Mono-objetivo e Multiobjetivo	49
3.3.4 Operador de Dominância de Pareto	50
3.4 Algoritmos Genéticos Multiobjetivo	51
3.4.1 <i>Nondominated Sorting Genetic Algorithm II (NSGA-II)</i>	52
3.5 Sistemas <i>Fuzzy</i> Genéticos	55
3.5.1 Adaptação Genética	57
3.5.2 Aprendizado Genético dos Componentes dos SFBR	58
3.6 Sistemas <i>Fuzzy</i> Genéticos Multiobjetivo	59
3.6.1 Adaptação Genética Multiobjetivo	61
3.6.2 Aprendizado Genético Multiobjetivo dos Componentes dos SFBR	62
3.7 Considerações finais	63
 CAPÍTULO 4 – GERAÇÃO GENÉTICA MULTIOBJETIVO DE SFBR COM FOCO NA DISTRIBUIÇÃO DAS SOLUÇÕES	 64
4.1 Considerações Iniciais	64
4.2 Metodologia para Geração de Base de Conhecimento <i>Fuzzy</i>	64

4.3	Cálculo do Espaçamento Ideal entre as Soluções	67
4.4	Índice de Distribuição	68
4.5	Escolha da Solução Central	72
4.6	<i>Non-Dominated Sorting Genetic Algorithm Distance Oriented</i> (NSGA-DO) . .	74
4.7	Geração de Base de Conhecimento <i>Fuzzy</i> usando NSGA-DO	77
4.7.1	Seleção de Regras <i>Fuzzy</i>	78
4.7.2	Otimização de Base de Dados	79
4.8	Considerações Finais	81
CAPÍTULO 5 – EXPERIMENTOS E ANÁLISE DOS RESULTADOS		82
5.1	Considerações Iniciais	82
5.2	Descrição dos Conjuntos de Dados	82
5.3	Experimentos Realizados	83
5.4	Considerações Finais	96
CAPÍTULO 6 – CONCLUSÕES E TRABALHOS FUTUROS		98
6.1	Considerações Iniciais	98
6.2	Contribuições	98
6.3	Limitações e Trabalhos Futuros	99
REFERÊNCIAS		101

Capítulo 1

INTRODUÇÃO

As metodologias da Inteligência Computacional (IC), particularmente os Sistemas *Fuzzy* (SF), têm provado ser ferramentas importantes para a modelagem de sistemas para os quais, devido à complexidade ou imprecisão, as abordagens clássicas não são adequadas. A bem sucedida utilização dos SF gerou a necessidade de investigação de metodologias complementares que possibilitassem o projeto automático de tais sistemas. Os crescentes esforços empregados nas duas últimas décadas para explorar o uso de algoritmos genéticos no projeto de SF deram origem ao campo dos Sistemas *Fuzzy* Genéticos (SFG), que têm atraído a atenção de pesquisadores de diferentes áreas (HERRERA, 2008; ZHANG; WANG; ZHANG, 2009; CORDÓN, 2011).

A área de SFG é considerada uma área de pesquisa bem estabelecida e bastante ativa, com um volume muito grande de publicações sendo geradas em todo o mundo. Importantes contribuições podem ser encontradas em diversos trabalhos sobre o assunto, como (CORDÓN, 2011; NOJIMA; ISHIBUCHI, 2009; STAVRAKOUDIS; THEOCHARIS, 2009a, 2009b; CORDÓN et al., 2007).

Um dos SF amplamente usados são os Sistemas *Fuzzy* Baseados em Regras (SFBR), os quais são, geralmente, constituídos de um Mecanismo de Inferência (MI) e de uma Base de Conhecimento (BC), que por sua vez é composta pela Base de Regras (BR) e pela Base de Dados (BD). Dentre estes SFBR encontram-se os Sistemas de Classificação *Fuzzy* (SCF), que são SFBR projetados com a finalidade específica de realizar a tarefa de classificação. Essa subclasse dos SFBR é uma das áreas que mais se beneficiam das vantagens do uso de conjuntos *fuzzy*, como trabalhar com informações imprecisas e modelar o problema de forma compreensível ao ser humano.

Um dos desafios existentes na modelagem dos SCF é a construção da BC. O número de regras, bem como o formato das regras que compõem a sua BR têm impacto direto no desempenho dos SCF. Quanto maior for o número de regras da BR dos SCF, maior será a sua acurácia

e menor será a sua interpretabilidade. Por outro lado, quanto menor for o número de regras da BR dos SCF, menor será a sua acurácia e maior será a sua interpretabilidade.

Outro fator que impacta diretamente o desempenho dos SCF é o formato dos conjuntos *fuzzy* que compõem a sua BD. Dependendo de como estes conjuntos foram ajustados, a acurácia e a interpretabilidade podem variar.

A partir deste contexto, a motivação, a hipótese e o objetivo que norteiam este doutorado são apresentados a seguir.

1.1 Motivações, hipótese e objetivo

A busca por SFBR com alta acuidade e boa interpretabilidade tem sido o foco de muitas pesquisas nos últimos anos. Uma vez que o aumento da acuidade e o aumento da interpretabilidade são objetivos conflitantes, os Algoritmos Genéticos Multiobjetivos (AGMO) têm sido amplamente empregados no esforço de balancear estes dois objetivos no processo de geração de regras *fuzzy* e de ajuste do formato dos conjuntos *fuzzy* (ISHIBUCHI; NOJIMA, 2007; SáEZ; COR-TÉS; nEZ, 2008; GACTO; ALCALÁ; HERRERA, 2008; ALCALÁ et al., 2009b; AGUILAR-LASSERRE et al., 2009; NOJIMA; ISHIBUCHI, 2009; PULKKINEN, 2009; KAYLANI et al., 2010; ISHIBUCHI; NAKASHIMA; NOJIMA, 2011; FAZZOLARI et al., 2013).

A otimização multiobjetivo por meio de AGMO fornece soluções não dominadas que nem sempre equilibram os objetivos devido à baixa diversidade entre estas soluções, fazendo com que as soluções encontradas beneficiem um dos objetivos. Com isso, um grande esforço vem sendo feito com o intuito de garantir a diversidade das soluções e, conseqüentemente, obter uma fronteira de Pareto cujas soluções estejam bem distribuídas (ADRA; FLEMING, 2011; HáJEK; SZöLLöS; VSÍSTEK, 2010; YANG et al., 2011). Entretanto, nos trabalhos encontrados na literatura, a distribuição das soluções ao longo da fronteira de Pareto é obtida tendo por base a diversidade da população total de indivíduos sem considerar de forma isolada a distribuição das soluções ao longo da fronteira de Pareto a cada geração do AGMO (HáJEK; SZöLLöS; VSÍSTEK, 2010; ZADOROJNIY et al., 2012; LI; YANG; LIU, 2014; AZEVEDO; ARAUJO, 2011).

Uma vez que o foco destas abordagens está na diversidade da população como um todo, a distribuição das soluções ao longo da fronteira de Pareto nem sempre será uma distribuição uniforme. Conseqüentemente, isso permite que haja um desbalanceamento entre as soluções localizadas na fronteira de Pareto.

Desta maneira, dado ao fato de que as abordagens encontradas na literatura não conside-

ram o posicionamento de cada solução ao longo da fronteira de Pareto como um critério de avaliação do balanceamento das soluções encontradas, este doutorado faz uso deste ambiente motivacional e é dirigido pela seguinte hipótese:

Considerar a distância entre as soluções presentes na fronteira de Pareto, durante o processo evolutivo de um AGMO, como uma medida de avaliação da qualidade do balanceamento entre acuidade e interpretabilidade, fará com que este AGMO convirja para uma fronteira de Pareto cujas soluções estejam distribuídas de uma maneira próxima a uma distribuição uniforme ao longo da fronteira de Pareto.

Tendo por base esta hipótese, este doutorado tem como principal objetivo:

Criar um método que permita a um AGMO encontrar soluções com alto grau de diversidade na fronteira de Pareto. O AGMO será usado na geração automática de um SCF, mantendo o equilíbrio entre a acuidade e a interpretabilidade.

Dados esta hipótese e objetivo, este doutorado apresenta como contribuição principal um método que permite a geração de um SCF com alto balanceamento entre a acurácia e a interpretabilidade. Partindo-se desta contribuição, outras contribuições específicas são apresentadas a seguir.

1.2 Contribuições

De forma abrangente, a inovação de pesquisa obtida por meio deste trabalho consiste no estudo, proposta, desenvolvimento e avaliação de um método capaz de gerar SCF a partir de dados numéricos, mantendo o balanceamento entre a acurácia e a interpretabilidade. Neste sentido, três contribuições específicas foram obtidas neste doutorado.

A primeira contribuição consiste da proposta e desenvolvimento de um índice que reflita o balanceamento das soluções observando a distribuição das soluções ao longo da fronteira de Pareto. Este índice não leva em consideração a diversidade da população, mas sim a distância entre as soluções presentes na fronteira de Pareto. Quanto mais uniforme é a distribuição das soluções ao longo da fronteira de Pareto, menor é o valor obtido pelo índice.

A segunda contribuição consiste da proposta e desenvolvimento de um método para a escolha eficiente de uma solução final presente na fronteira de Pareto. Uma vez que os AGMOs geram mais de uma solução factível, é importante dispor de um método que auxilie na escolha da solução que reflete o maior equilíbrio entre os objetivos que estão sendo otimizados.

Por fim, uma terceira contribuição consiste da proposta e desenvolvimento de um AGMO

capaz de gerar SCF com boa acurácia e interpretabilidade. Para tanto, este AGMO utiliza em seu operador de seleção o índice desenvolvido nesta tese. O intuito é beneficiar as soluções que melhor se distribuem ao longo da fronteira de Pareto de cada geração genética.

Cada uma destas três contribuições são apresentadas e discutidas ao longo desta tese, a qual é organizada conforme apresentado na próxima seção.

1.3 Organização da Tese

Para mostrar a validação da hipótese e atingir o objetivo proposto neste doutorado, esta tese está dividida nos seguintes capítulos:

- No Capítulo 2 são apresentados os conceitos relacionados aos sistemas *fuzzy*, bem como aspectos relacionados à interpretabilidade;
- O Capítulo 3 apresenta temas da Computação Evolutiva relevantes para este trabalho, tais como Algoritmos Genéticos e Algoritmos Genéticos Multiobjetivos;
- No Capítulo 4 são apresentados o método proposto para a geração de SFBR, o algoritmo NSGA-DO, bem como a metodologia em que o algoritmo proposto é aplicado para a geração de Classificadores *Fuzzy*;
- No Capítulo 5 são apresentados os experimentos realizados e os resultados obtidos;
- Por fim, no Capítulo 6 são apresentadas as conclusões, as principais contribuições e limitações da proposta desenvolvida.

Capítulo 2

SISTEMAS *Fuzzy*

2.1 Considerações Iniciais

Os Sistemas *Fuzzy* (SF) podem ser definidos, de maneira bastante geral, como sistemas que utilizam a teoria dos conjuntos *fuzzy* proposta por Zadeh (1965) com o objetivo de propor um formalismo capaz de capturar, representar e processar dados associados a noções linguísticas ou objetos com fronteiras imprecisas. O resultado foi a formalização de uma plataforma de processamento de informações imprecisas inspirada no conhecimento humano. Desta maneira os sistemas *fuzzy* representam uma possível ponte entre o processamento linguístico e o processamento numérico (DUBOIS; PRADE, 1998).

A lógica *fuzzy* pode ser entendida como uma extensão da lógica binária. Na lógica binária uma proposição sempre assume os valores-verdade verdadeiro ou falso. A lógica *fuzzy*, em contrapartida, assume que os valores-verdade são conjuntos *fuzzy*, permitindo assim a utilização de um raciocínio aproximado que pode ser linguisticamente interpretável.

Nas próximas seções são discutidos os fundamentos da teoria de conjuntos *fuzzy*, da lógica *fuzzy* e dos sistemas *fuzzy*.

2.2 Conjuntos *Fuzzy*

Os conjuntos clássicos (*crisp sets*) são amplamente usados na matemática e na ciência da computação. Entretanto, este tipo de conjunto não é adequado para representar a natureza do pensamento humano, que tende a ser impreciso e abstrato (JANG; SUN; MIZUTANI, 1997a).

Em um conjunto clássico um elemento pertence completamente ao conjunto ou não pertence ao conjunto. Todavia, o mundo real apresenta situações em que os conjuntos clássicos

não são capazes de definir de forma adequada certas classes de elementos por não possuírem flexibilidade suficiente, fazendo com que ocorra uma transição abrupta entre a pertinência absoluta e a não pertinência absoluta.

Já os conjuntos *fuzzy*, considerados uma generalização dos conjuntos clássicos, possibilitam a transição gradual de conjunto, permitindo que cada elemento pertença parcialmente a um ou mais conjuntos.

2.2.1 Funções de Pertinência

Um dos conceitos mais importantes da teoria de conjuntos *fuzzy* é o conceito de função de pertinência. Define-se uma função de pertinência como um mapeamento como explicitado em 2.1.

$$\mu_A : \mathbf{X} \rightarrow [0, 1] \quad (2.1)$$

em que \mathbf{X} um conjunto de objetos, denominado universo de discurso, e x um elemento de \mathbf{X} .

Um conjunto *fuzzy* A é definido como:

$$A = \{(x, \mu_A(x)) \mid x \in \mathbf{X}\} \quad (2.2)$$

em que $\mu_A(x)$ é uma função de pertinência que indica o grau de pertinência de um objeto x ao conjunto *fuzzy* A . Assim $\mu_A(x) = 1$ indica que o elemento pertence totalmente ao conjunto *fuzzy* A e $\mu_A(x) = 0$ indica que o elemento não pertence ao conjunto *fuzzy* A . O grau com que x pertence a A é dado pela função de pertinência $\mu_A(x)$.

Os formatos mais comuns de funções de pertinência são: triangular, trapezoidal e Gaussiana (KLIR, 1995).

Função triangular: A função triangular pode ser apresentada na forma parametrizada como:

$$\mu_A(x) = \begin{cases} 0, & \text{se } x \leq a, \\ \frac{x-a}{b-a}, & \text{se } a < x \leq b, \\ \frac{c-x}{c-b}, & \text{se } b < x \leq c, \\ 0, & \text{se } k > c \end{cases} \quad (2.3)$$

Alternativamente, pode-se reescrever a Equação 2.3 como:

$$\mu_A(x) = \max \left(\min \left(\frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right) \quad (2.4)$$

em que b é o valor modal de μ , ou seja, o maior valor de μ . Os pontos a e c representam os limites inferior e superior dos elementos compatíveis com o conjunto *fuzzy*, respectivamente. Graficamente, pode-se interpretar o conjunto $\{a, b, c\}$ como as coordenadas dos três vértices do triângulo formado pela função de pertinência.

Função trapezoidal: A função trapezoidal é especificada por um conjunto de quatro parâmetros, $\{a, b, c, d\}$, conforme a Equação 2.5:

$$\mu_A(x) = \begin{cases} 0, & \text{se } x \leq a, \\ \frac{x-a}{b-a}, & \text{se } a < x \leq b, \\ 1, & \text{se } b < x \leq c \\ \frac{d-x}{d-c}, & \text{se } c < x \leq d, \\ 0, & \text{se } x > d \end{cases} \quad (2.5)$$

Alternativamente, pode-se reescrever a Equação 2.5 como:

$$\mu_A(x) = \max \left(\min \left(\frac{x-a}{b-a}, \frac{d-x}{d-b} \right), 0 \right) \quad (2.6)$$

Graficamente, pode-se interpretar o conjunto $\{a, b, c, d\}$ como as coordenadas dos quatro vértices do trapézio formado pela função de pertinência. Para o caso especial em que $b=c$ a função trapezoidal se reduz a uma função triangular. Para o caso especial $a=b$ e $c=d$ tem-se um conjunto clássico.

Função Gaussiana: Uma função de pertinência Gaussiana é especificada por dois parâmetros, a média (ω) e a abertura (σ), conforme a Equação 2.7.

$$\mu_A = e^{-\frac{1}{2} \left(\frac{x-\omega}{\sigma} \right)^2} \quad (2.7)$$

A função Gaussiana é simétrica ao redor de ω e sua abertura é especificada pelo parâmetro σ .

Devido à sua formulação simples e eficiência computacional, as funções do tipo triangular e trapezoidal são amplamente utilizadas, especialmente em implementações com requisitos de funcionamento em tempo real (KAYA; ALHAJJ, 2003).

A especificação do formato de uma função de pertinência nem sempre é óbvia, podendo

inclusive não estar ao alcance do conhecimento de um especialista para a aplicação desejada.

No entanto, existem conjuntos *fuzzy* cujos parâmetros das funções de pertinência são completamente definidos pelo especialista. Nestes casos, a escolha de funções triangulares e trapezoidais é mais comum porque a ideia de definição de regiões de pertinência total, média e nula é mais intuitiva que a especificação de valor modal e dispersão, associados ao projeto de funções Gaussianas. Em trabalhos mais recentes existe a tendência ao projeto automático de sistemas *fuzzy* nos quais os parâmetros das funções de pertinência são ajustados no sentido de otimizar alguma função objetivo.

Em geral essa otimização é realizada a partir de um conjunto significativo de dados de treinamento, ou seja, um conjunto de dados que descreva adequadamente o comportamento do sistema.

2.2.2 Operações Básicas sobre Conjuntos *Fuzzy*

As operações sobre conjuntos são essenciais para a representação e processamento do conhecimento. A seguir são descritas as operações *fuzzy* de complemento, interseção e união.

Complemento *Fuzzy*

Se A é um conjunto *fuzzy* em \mathbf{X} , o complemento *fuzzy* de A é um conjunto *fuzzy* denotado por cA e o valor de $c\mu_A(x)$ para $x \in \mathbf{X}$ é interpretado como o grau com que x pertence a cA e pode ser definido pela função:

$$c : [0, 1] \rightarrow [0, 1] \quad (2.8)$$

que atribui o valor $c(\mu_A(x))$ a cada grau de pertinência $\mu_A(x)$ de qualquer conjunto *fuzzy* A . O valor $c(\mu_A(x))$ é interpretado como o valor $cA(x)$, ou seja, para todo $x \in \mathbf{X}$, por definição, $c(A(x)) = cA(x)$.

Dado um conjunto *fuzzy* A , o complemento *fuzzy* cA é obtido por meio da aplicação da função c aos valores $\mu_A(x)$ para todo $x \in X$.

Para que os complementos *fuzzy* obtidos possam ser significativos e possam efetivamente expressar a ideia de complemento, é importante que a função c satisfaça os seguintes axiomas:

- Condições limites: $c(0) = 1$ e $c(1) = 0$
- Monotonicidade: $\forall a, b, c \in [0, 1]$ se $a \leq b$ então $c(a) \geq c(b)$.

Todas as funções que satisfaçam esses axiomas formam a classe geral de complementos *fuzzy*. O primeiro axioma garante que os complementos *fuzzy* podem ser utilizados para conjuntos clássicos. O segundo axioma garante que um aumento do grau de pertinência de um elemento a em um conjunto *fuzzy* A implica uma diminuição da pertinência desse elemento ao complemento.

Um requisito opcional, que não costuma ser satisfeito pela maioria dos complementos *fuzzy*, é a condição de involução da negação *fuzzy* $c(c(a)) = a$, que garante que a negação dupla do conjunto é o próprio conjunto.

Os tipos mais comuns de negação *fuzzy* encontrados na literatura (KLIR, 1995) são:

- Complemento: $c_1(a) = 1 - a$
- Sugeno: $c_2(a, s) = \frac{1-a}{1+s \cdot a}$, $s > -1$
- Yager: $c_3(a, w)^{\frac{1}{w}} = (1 - a^w)^{\frac{1}{w}}$, $w > 0$

Podemos observar que $c_2(a, 0) = c_1(a)$ e também $c_3(a, 1) = c_1(a)$.

Interseção (t-normas)

A interseção de dois conjuntos *fuzzy* é definida pela operação binária $t : [0, 1]^2 \rightarrow [0, 1]$ que associa um valor de pertinência a um par de valores de pertinência.

Sejam A e B dois conjuntos *fuzzy* definidos no universo de discurso X , e $\mu_A(x)$ e $\mu_B(x)$ suas funções de pertinência, respectivamente, então:

$$\mu_{A \cap B}(x) = \mu_A(x) t \mu_B(x) \quad (2.9)$$

De maneira semelhante ao complemento *fuzzy*, para que uma função t assim definida possa ser considerada e adotada como interseção *fuzzy*, deve satisfazer às seguintes propriedades que garantem que os conjuntos *fuzzy* resultantes da aplicação de t sejam significativos como interseções *fuzzy*, para todo $x, y, z \in [0, 1]$:

- Condição limite: $0 t a = 0$ e $1 t a = a$
- Monotonicidade: Se $a \leq b$, então $c t a \leq c t b$
- Comutatividade: $a t b = b t a$
- Associatividade: $a t (b t c) = (a t b) t c$

As funções conhecidas como *t-normas* possuem tais propriedades e, geralmente, são adotadas como intersecções *fuzzy*. Algumas das *t-normas* mais utilizados na literatura (KLIR, 1995) são:

- mínimo: $atb = \min\{a, b\}$
- produto algébrico: $atb = ab$
- produto restrito: $atb = \max\{0, a + b - 1\}$
- produto drástico: $atb = \begin{cases} a, & \text{se } b = 1, \\ b, & \text{se } a = 1, \\ c, & \text{se } a e b = 1 \end{cases}$
- Schweizer e Sklar: $atb = [\max\{0, (a^{-p} + b^{-p} - 1)\}]^{\frac{-1}{p}}, p > 0$
- Yager: $atb = 1 - \min\{1, [(1 - a)^q + (1 - b)^q]^{\frac{1}{q}}\}, q > 0$
- Dubois e Prade: $atb = \frac{ab}{\max\{a, b, \alpha\}}, \alpha \in [0, 1]$
- Hamacher: $atb = \frac{ab}{\gamma + (1 - \gamma)(a + b - ab)}, \gamma > 0$
- Frank: $atb = \log_s \left[1 + \frac{(s^a - 1)(s^b - 1)}{s - 1} \right], s > 0$
- Sugeno: $atb = \max\{0, (\lambda + 1)(a + b - 1) - \lambda \cdot a \cdot b\}, \lambda \geq 1$
- Dombi: $\frac{1}{1 + [(a^{-1} - 1)^\lambda + (b^{-1} - 1)^\lambda]^{\frac{1}{\lambda}}}, \lambda > 0$

União (s-normas)

De maneira semelhante à intersecção *fuzzy*, a união *fuzzy* de dois conjuntos *fuzzy* é definida pela operação binária $s : [0, 1]^2 \rightarrow [0, 1]$ que associa um valor de pertinência a um par ordenado de valores de pertinência. Se A e B são os conjuntos *fuzzy* em questão e \mathbf{X} o universo de discurso em que estão definidos, para todo $x \in \mathbf{X}$ temos:

$$\mu_{A \cup B}(x) = \mu_A(x) s \mu_B(x) \quad (2.10)$$

De maneira semelhante ao complemento e à intersecção *fuzzy*, para que uma função s , assim definida, possa ser considerada e adotada como união *fuzzy*, ela deve satisfazer propriedades que garantam que os conjuntos *fuzzy* resultantes da aplicação de s sejam significativos como uniões *fuzzy*. As funções conhecidas como *t-conormas*, ou *s-normas*, possuem tais propriedades e, geralmente, são adotadas como uniões *fuzzy*. As propriedades das *s-normas* citadas são:

- Condição limite: $0sx = x$ e $1sx = 1$
- Monotonicidade: Se $x \leq y$ e $w \leq z$, então $xsw \leq ysz$
- Comutatividade: $xsy = ysx$
- Associatividade: $xs(ysz) = (xsy)sz$

Os tipos de s-normas triangulares mais utilizados na literatura (KLIR, 1995) são:

- máximo: $asb = \max\{a, b\}$
- soma algébrica: $asb = a + b - ab$
- soma restrita: $asb = \max\{1, a + b\}$
- soma drástica: $asb = \begin{cases} a, & \text{se } b = 0, \\ b, & \text{se } a = 0, \\ 1, & \text{se } a \text{ e } b = 0 \end{cases}$
- Schweizer e Sklar: $asb = 1 - [\max\{0, ((1-a)^{-p} + (1-b)^{-p} - 1)\}]^{\frac{1}{p}}$, $p > 0$
- Yager: $asb = \min\{1, [a^q + b^q]^{\frac{1}{q}}\}$, $q > 0$
- Dubois e Prade: $asb = \left[a + b - ab - \frac{\min\{a, b, (1-\alpha)\}}{\max\{1-a, 1-b, \alpha\}} \right]$, $\alpha \in [0, 1]$
- Hamacher: $asb = \frac{a+b-(\gamma-2)ab}{1+(\gamma-1)ab}$, $\gamma > 0$
- Frank: $asb = 1 - \log_s \left[1 + \frac{(s^{1-a}-1)(s^{1-b}-1)}{s-1} \right]$, $s > 0$
- Sugeno: $asb = \min\{1, a + b - \gamma ab\}$, $\gamma \geq -1$
- Dombi: $\frac{1}{1+[(a^{-1}-1)^{-\lambda} + (b^{-1}-1)^{-\lambda}]^{\frac{1}{\lambda}}}$, $\lambda > 0$

2.2.3 Relações Fuzzy

A relação fuzzy binária $R(\mathbf{X}, \mathbf{Y})$, onde \mathbf{X} e \mathbf{Y} são conjuntos clássicos (*crisp*), é um conjunto fuzzy caracterizado pela função de pertinência $\mu_R(x, y)$ no qual $x \in \mathbf{X}$ e $y \in \mathbf{Y}$ e $\mu_R(x, y)$ é um valor no intervalo $[0, 1]$ que define o grau com que x e y estão relacionados pela relação R .

Relação Fuzzy Binária

Sejam \mathbf{X} e \mathbf{Y} dois universos de discurso. Pode-se definir o conjunto fuzzy R como:

$$R : \mathbf{X} \times \mathbf{Y} \rightarrow [0, 1] \quad (2.11)$$

$$R : \{(x, y), \mu_R(x, y) \mid (x, y) \in \mathbf{X} \times \mathbf{Y}\} \quad (2.12)$$

a Equação 2.12 é uma relação *fuzzy* binária no universo de discurso. Por exemplo, seja $\mathbf{X} \equiv \mathbf{Y} \equiv \mathbb{R}^+$ e $R = "x \text{ é muito menor que } y"$. Supondo que a função de pertinência de R possa ser aproximada por:

$$\mu_R(x, y) = \begin{cases} \frac{y-x}{x+y-2}, & \text{se } y > x, \\ 0, & \text{se } y \leq x \end{cases} \quad (2.13)$$

Para os universos de discurso $\mathbf{X} = \{3, 4, 5\}$ e $\mathbf{Y} = \{3, 4, 5, 6, 7\}$, pode-se expressar a mesma relação *fuzzy* R por meio de uma matriz de relação:

$$R = \begin{bmatrix} 0 & 0,200 & 0,333 & 0,429 & 0,500 \\ 0 & 0 & 0,143 & 0,250 & 0,333 \\ 0 & 0 & 0 & 0,111 & 0,200 \end{bmatrix} \quad (2.14)$$

no qual o elemento na linha i e coluna j indica o grau de pertinência ao conjunto "*muito maior*" do elemento (i, j) sendo $i \in \mathbf{X}$ e $j \in \mathbf{Y}$.

2.2.4 Composição de Relações Fuzzy

As relações *fuzzy* definidas em produtos cartesianos distintos podem ser combinadas utilizando operadores de composição. Existem diversos operadores de composição possíveis, sendo que o mais conhecido é a composição max-min proposta por (ZADEH, 1965).

Composição Max-min

Sejam R_1 e R_2 duas relações *fuzzy* definidas em $\mathbf{X}_1 \times \mathbf{X}_2$ e $\mathbf{X}_2 \times \mathbf{X}_3$ respectivamente. A composição Max-min de R_1 e R_2 é um conjunto *fuzzy* definido por:

$$R_1 \circ R_2 = \{(x_1, x_3), \mu_{R_1 \circ R_2}(x_1, x_3) \mid x_1 \in \mathbf{X}_1, x_2 \in \mathbf{X}_2, x_3 \in \mathbf{X}_3\} \quad (2.15)$$

em que:

$$\mu_{R_1 \circ R_2}(x_1, x_3) = \max_{x_2 \in X_2} \min(\mu_{R_1}(x_1, x_2), \mu_{R_2}(x_2, x_3)) \quad (2.16)$$

Quando as relações R_1 e R_2 são expressas como matrizes de relação, o cálculo de $R_1 \circ R_2$ é semelhante a uma multiplicação matricial, exceto pelo fato que os operadores de multiplicação e soma são substituídos por min e max respectivamente. A composição Max-prod foi proposta como uma alternativa à composição Max-min (JANG; SUN; MIZUTANI, 1997a), para alcançar maior tratabilidade matemática.

Composição Max-Prod

Assumindo a mesma notação utilizada na definição da composição Max-min, pode-se definir a composição Max-prod como:

$$\mu_{R_1 \circ R_2}(x_1, x_3) = \max_{x_2 \in X_2} \{\mu_{R_1}(x_1, x_2) \cdot \mu_{R_2}(x_2, x_3)\} \quad (2.17)$$

Composição genérica

Analogamente pode-se utilizar o mesmo raciocínio para definir uma composição através de quaisquer pares de s -normas e t -normas (NGUYEN; GANESH; GONG, 1998).

$$\mu_{R_1 \oplus R_2}(x_1, x_3) = s_{x_2 \in X_2} [\mu_{R_1}(x_1, x_2) t \mu_{R_2}(x_2, x_3)] \quad (2.18)$$

ou ainda:

$$\mu_{R_1 \oplus R_2}(x_1, x_3) = t_{x_2 \in X_2} [\mu_{R_1}(x_1, x_2) s \mu_{R_2}(x_2, x_3)] \quad (2.19)$$

2.3 Regras Fuzzy

Regras fornecem uma maneira formal de representar diretrizes e estratégias, sendo muitas vezes apropriadas quando o conhecimento do domínio resulta da experiência ou de associações empíricas. Sistemas baseados em regras são construídos sobre um conjunto de regras e usam uma coleção de fatos para fazer inferências.

Regras são úteis para a representação de conhecimento sobre um determinado domínio. A modelagem de sentenças em linguagem natural ou artificial é relativamente simples por meio de regras *fuzzy* (JANG; SUN; MIZUTANI, 1997b). Além disso, as regras permitem a representação do

conhecimento impreciso, o que facilita a modelagem de problemas do mundo real. As regras, *fuzzy* ou não, podem ser usadas por sistemas computacionais em seus mecanismos de raciocínio para inferir conclusões a partir de fatos conhecidos.

A próxima seção descreve os conceitos de variáveis e regras *fuzzy*, bem como os fundamentos dos processos de inferência que utilizam regras *fuzzy*.

2.3.1 Variáveis Linguísticas

Conforme apontado por Zadeh, técnicas convencionais para análise de sistemas são essencialmente inadequadas para o tratamento de sistemas baseados no conhecimento humano, cujo comportamento é influenciado pela percepção, julgamento e emoções. Essa é uma manifestação do princípio da incompatibilidade: “Com o aumento da complexidade do sistema, nossa habilidade de realizar indicações precisas e significativas sobre seu funcionamento diminui até ser alcançado um nível em que precisão e significado se tornam características quase que mutuamente exclusivas” (ZADEH, 1973). Isso motivou a proposta do conceito de variáveis linguísticas como uma alternativa na modelagem do pensamento humano em que a informação é processada através de conjuntos *fuzzy*. Formalmente uma variável linguística é caracterizada por uma quintupla:

$$\langle \chi, T, \mathbf{X}, G, M \rangle \quad (2.20)$$

na qual:

- χ é o nome da variável linguística;
- T é o conjunto de rótulos linguísticos primários associados a χ . Cada elemento de T representa um rótulo dos termos que a variável χ pode assumir;
- \mathbf{X} é o universo de discurso da variável linguística χ ;
- G é a gramática para geração dos rótulos;
- M é a regra que associa cada rótulo a um conjunto *fuzzy* no universo \mathbf{X} , representando o seu significado.

Por exemplo, seja a variável linguística velocidade ($\chi = \textit{velocidade}$) no universo de discurso $\mathbf{X} = [0, 150]$ e uma variável base x . Um conjunto de termos linguísticos associados à

variável *velocidade* poderia ser: $T(\textit{velocidade}) = \{\textit{baixa}, \textit{média}, \textit{alta}\}$. Seja $P(\mathbf{X})$ uma coleção de conjuntos *fuzzy* definidos no universo \mathbf{X} . A semântica de uma variável linguística define um mapeamento:

$$M : T(\chi) \rightarrow P(\mathbf{X}) \quad (2.21)$$

que estabelece uma relação entre o conjunto de termos $T(\chi)$ a uma partição *fuzzy* P .

A gramática G define como os termos linguísticos primários $\{\textit{baixa}, \textit{média}, \textit{alta}\}$ serão associados aos modificadores $\{\textit{muito}, \textit{pouco}, \textit{maior}, \textit{menor}, \textit{ou}, \textit{não}\}$ para formar os nomes dos termos não-primários. Usualmente as funções de pertinência associadas aos termos linguísticos primários têm formatos conhecidos. Os conjuntos *fuzzy* associados aos termos não primários, em contrapartida, podem ser obtidos através de modificadores pré-especificados (PEDRYCZ; GOMIDE, 1998).

O processo de divisão de um domínio de uma variável em conjuntos *fuzzy* define a chamada partição *fuzzy*. Uma partição *fuzzy* pode ser uniforme (os termos linguísticos têm funções de pertinência de formato semelhante, transladadas e equidistantes) ou não-uniforme. A granularidade da partição é definida pelo número de termos linguísticos. Um número baixo de termos linguísticos define uma partição esparsa ou grossa, enquanto um número alto resulta em uma partição fina.

Não existe uma metodologia consistente para a determinação da partição *fuzzy* ideal. Em geral essa tarefa é realizada manualmente através da intervenção de um especialista ou utilizando um método de particionamento automático a partir de dados de treinamento. Na ausência de um especialista ou de abordagens de ajuste automático, partições utilizando de 5 a 7 termos linguísticos uniformemente distribuídos, são utilizadas com frequência (PEDRYCZ; GOMIDE, 1998).

2.3.2 Sintaxe e Semântica das Regras Fuzzy

As regras *fuzzy* são também conhecidas como implicações *fuzzy* ou declarações condicionais *fuzzy*. Elas são apropriadas quando o conhecimento do problema é resultado de associações empíricas e experiências de um especialista, ou quando se deseja uma representação linguística do conhecimento adquirido. Em geral, as regras *fuzzy* assumem a forma:

$$\textit{Se } x \textit{ é } A \textit{ então } y \textit{ é } B \quad (2.22)$$

em que A e B são rótulos (ou termos) linguísticos de conjuntos *fuzzy* definidos nos conjuntos base \mathbf{X} e \mathbf{Y} , respectivamente, e x e y são variáveis linguísticas. Frequentemente “ x é A ” é denominado de antecedente ou premissa, enquanto “ y é B ” é denominado de conseqüente ou conclusão (JANG; SUN; MIZUTANI, 1997a).

A regra *fuzzy* mostrada na equação 2.22 pode ser abreviada por $A \rightarrow B$ e pode ser definida como uma relação *fuzzy* R no produto cartesiano $\mathbf{X} \times \mathbf{Y}$, onde \mathbf{X} e \mathbf{Y} são conjuntos convencionais. Seja μ_R a função de pertinência associada à relação *fuzzy* R , a semântica da regra *fuzzy* pode ser especificada através de uma função $f : [0, 1] \times [0, 1] \rightarrow [0, 1]$:

$$\mu_R(x, y) = f(\mu_A(x), \mu_B(y)) \quad (2.23)$$

De acordo com Pedrycz e Gomide (1998), as relações *fuzzy* induzidas por regras *fuzzy* são derivadas de três classes principais de funções: conjunções *fuzzy*, disjunções *fuzzy* e implicações *fuzzy*, sendo as conjunções e implicações as mais comuns. As conjunções e as disjunções podem ser vistas como generalizações duais do produto cartesiano *fuzzy* através de t -normas e s -normas, enquanto implicações *fuzzy* representam generalizações das implicações da lógica multi-valorada de Lukasiewicz (1970).

2.4 Sistemas de Inferência Fuzzy

As regras *fuzzy* são ferramentas eficientes na modelagem de sentenças em linguagem natural (JANG; SUN; MIZUTANI, 1997a). Esta modelagem utiliza o conceito de variáveis linguísticas para gerar regras *fuzzy* que são mapeadas através de relações *fuzzy*, permitindo a investigação de diferentes esquemas de raciocínio aproximado. Nestes esquemas, procedimentos de inferência baseados no conceito de regra composicional de inferência são utilizados para derivar conclusões, a partir de um conjunto de regras e fatos conhecidos.

Um sistema de inferência *fuzzy* é uma ferramenta de computação baseada nos conceitos de conjuntos *fuzzy*, regras *fuzzy* e raciocínio *fuzzy*. Tais sistemas são utilizados em diversas áreas, tais como controle automático, classificação de dados, tomada de decisões, predição de séries temporais, robótica e reconhecimento de padrões (JANG; SUN; MIZUTANI, 1997a). Outras nomenclaturas para os sistemas *fuzzy* encontradas na literatura incluem: Sistemas Fuzzy Baseados em Regras (SFBR), Sistemas Especialistas Fuzzy (KANDEL, 1992), Modelos Fuzzy (TAKAGI; SUGENO, 1985; SUGENO; KANG, 1988), Memórias Associativas Fuzzy (KOSKO, 1992) e Controlador Lógico *fuzzy* (KANDEL, 1992; LEE, 1990b, 1990a).

A estrutura básica de um sistema de inferência *fuzzy* é constituída de dois componentes principais:

- Base de Conhecimento (BC), formada pela Base de Regras (BR), que contém um conjunto de regras *fuzzy*, e pela Base de Dados (BD), que define as funções de pertinência utilizadas nas regras *fuzzy*;
- Mecanismo de Inferência (MI), que executa o procedimento de inferência sobre as regras *fuzzy* e retorna fatos para formar uma saída aceitável ou conclusão.

Observa-se que o sistema básico de inferência *fuzzy* pode receber como entrada tanto conjuntos *fuzzy* quanto entradas não-*fuzzy* (que são interpretadas como conjuntos do tipo *singleton*). As saídas produzidas são geralmente conjuntos *fuzzy* ou valores numéricos, dependendo do modelo de inferência utilizado. Em alguns casos é necessário obter como saída um valor escalar, mesmo nos casos em que as regras produziram uma saída não-*fuzzy*. Portanto, é necessário um método de transformação da saída *fuzzy* em não-*fuzzy*, chamado de *defuzificador*, para obter um valor escalar que melhor represente o conjunto *fuzzy* obtido.

O tipo de sistema *fuzzy* utilizado ao longo desse trabalho, entretanto, não necessita de tal procedimento, o que por essa razão, não será profundamente detalhado.

Existem vários tipos de sistemas *fuzzy*. Na maioria dos casos, o antecedente é formado por proposições linguísticas e a distinção entre os modelos se dá no conseqüente das regras *fuzzy*. Entre os modelos mais conhecidos pode-se destacar:

Linguístico ou **Mamdani**: utiliza conjuntos *fuzzy* nos conseqüentes das regras *fuzzy* (MAMDANI; ASSILIAN, 1975). A saída final é representada por um conjunto *fuzzy* resultante da agregação da saída inferida de cada regra. Para se obter uma saída final não *fuzzy* adota-se um dos métodos de transformação da saída *fuzzy* em não *fuzzy* (JANG; SUN; MIZUTANI, 1997a).

Interpolativo ou **Takagi-Sugeno**: o conseqüente é representado por uma função das variáveis de entrada (TAKAGI; SUGENO, 1993). A saída final é obtida pela média ponderada das saídas inferidas de cada regra. Os coeficientes de ponderação são dados a partir do grau de ativação de cada regra.

Tsukamoto: utiliza funções de pertinência monotônicas no conseqüente (TSUKAMOTO, 1993). A saída do sistema é um valor não-*fuzzy* obtido da média ponderada das saídas inferidas de cada regra.

2.5 Sistema de Classificação Fuzzy

Sistemas de Classificação Fuzzy (SCF) são Sistemas Fuzzy Baseados em Regras projetados com a finalidade específica de realizar a tarefa de classificação. Assim, os SCF utilizam métodos de raciocínio próprios para essa tarefa.

O formato da regra fuzzy de classificação segue o formato geral das regras fuzzy descrito na Seção 2.4 e pode ser expresso por:

$$R_K : SE \ x_1 \text{ é } A_{K_1} \text{ e } x_2 \text{ é } A_{K_2}, ENT\tilde{A}O \ Classe = C_j \quad (2.24)$$

no qual R_K é o identificador da regra k , x_1, \dots, x_n são os atributos do padrão considerado no problema (representados aqui por variáveis linguísticas), A_{K_1}, \dots, A_{K_n} são os rótulos linguísticos usados para representar os valores de tais n atributos e C_j ($j = 1, \dots, M$) é a classe a que pertence o padrão.

Métodos de raciocínio fuzzy para classificação são procedimentos que permitem a inferência a partir dos consequentes de um conjunto de regras do tipo *SE...ENTÃO*, e um padrão de entrada, com o objetivo de classificar esse padrão. Os métodos mais utilizados são: Método de Raciocínio Fuzzy Clássico e Método de Raciocínio Fuzzy Geral.

O método de Raciocínio Fuzzy Clássico (CHI; YAN; PHAM, 1996; GONZÁLEZ; PÉREZ, 1999) classifica um padrão de entrada de acordo com a regra que possui maior grau de compatibilidade com o mesmo, conforme a definição a seguir.

Seja $e_p = (a_{p_1}, a_{p_2}, \dots, a_{p_n})$ um padrão a ser classificado, R_1, R_2, \dots, R_S um conjunto de S regras do SCF, cada uma com n antecedentes e L_j um subconjunto de R ($L_j \subseteq R$). Seja $A_{K_i}(a_{p_i})$, o grau de pertinência do atributo a_{p_i} no i -ésimo conjunto fuzzy da regra R_K .

O Método de Raciocínio Fuzzy Clássico segue as seguintes etapas para a classificação do padrão e_p :

1. Calcular o grau de compatibilidade entre o padrão e_p e cada regra R_K , para $K = 1, \dots, S$.

$$Compat(R_K, e_p) = t(A_{K_1}(a_{p_1}), A_{K_2}(a_{p_2}), \dots, A_{K_n}(a_{p_n}))$$

no qual t denota uma t -norma.

2. Encontrar a regra $R_{K_{\max}}$ que possui o maior grau de compatibilidade com o padrão.

$$Y_j = \max\{Compat(R_K, e_p)\}, k = 1, 2, \dots, S$$

3. Atribuir a classe C_h ao padrão e_p , onde C_h é a classe do consequente da regra R_k .

$$Y_h = \max_{j=1, \dots, M} Y_j$$

O método de Raciocínio *Fuzzy* Geral, que combina as informações providas por todas as regras de uma mesma classe para classificar o padrão, é definido a seguir para classificar o padrão e_p :

1. Calcular o grau de compatibilidade entre o padrão e_p e cada regra R_R , para $R = 1, \dots, S$.

$$Compat(R_K, e_p) = t(A_{K_1}(a_{p_1}), A_{K_2}(a_{p_2}), \dots, A_{K_n}(a_{p_n}))$$

no qual t denota uma t -norma.

2. Calcular para cada classe C o valor de $Classe_C$, ou seja, o grau de classificação do padrão na classe, agregando os graus de compatibilidade do passo anterior de todas as regras cuja classe predita é C :

$$Classe_C = f\{Compat(R_K, e_p) | C \text{ é a classe do consequente de } R_K\}$$

sendo f um operador de agregação tal que $\min \leq f \leq \max$

3. A classe C será atribuída ao padrão e_p , sendo C a classe correspondente ao valor máximo de $Classe_C$ encontrado no passo anterior.

2.6 Interpretabilidade de Sistemas *Fuzzy* Baseados em Regras

A interpretabilidade tem um papel importante na modelagem *fuzzy* (ZHOU; GAN, 2008), e é reconhecida como a principal vantagem dos sistemas *fuzzy* em relação aos sistemas clássicos como, por exemplo, as Redes Neurais Artificiais, que são vistas como caixas pretas, pois fórmulas matemáticas definem o mapeamento entre entradas e saídas, sendo assim difícil para o usuário entender o porquê de o sistema ter inferido algumas conclusões.

Por outro lado, as decisões tomadas pelos sistemas *fuzzy* são transparentes, já que todos os elementos do sistema podem ser verificados e compreendidos por um ser humano. Todavia, a interpretabilidade depende de vários fatores, tais como a estrutura do SFBR, o número de regras, o número de atributos, o número de rótulos linguísticos e o formato dos conjuntos *fuzzy*. Por esta razão, um grande esforço tem sido empregado pela comunidade científica para definir

medidas que possam ser utilizadas com o objetivo de se obter um índice que possa aferir quão interpretável é um SFBR.

Embora seja possível encontrar na literatura diversas tentativas de definir um índice de interpretabilidade, ainda não há um índice universal amplamente aceito (ZHOU; GAN, 2008).

O conceito de interpretabilidade aparece em muitas áreas como a educação, medicina e a ciência da computação, sob vários nomes como compreensibilidade, compreensão, inteligibilidade, transparência, legibilidade, etc.

Todos esses termos são geralmente considerados como sinônimos, o que poderia gerar alguma confusão. No entanto, alguns autores (MENCAR; FANELLI, 2008) fazem distinção entre o termo *transparência* (legibilidade), que diz respeito à estrutura do modelo e o termo *compreensibilidade* (compreensão), que está voltado aos aspectos cognitivos, pois está relacionado ao ser humano ou, mais especificamente, para sistemas humanísticos, definidos por Zadeh como aqueles cujo comportamento é fortemente influenciado pelo julgamento humano, percepção e emoções (ZADEH, 1965).

Pode-se observar que a legibilidade é considerada um pré-requisito para compreensibilidade. Por conta disto, em alguns trabalhos, o termo interpretabilidade é usado para se referir a ambos os termos: legibilidade e compreensibilidade.

Não é fácil encontrar na literatura uma definição formal para a interpretabilidade que tenha aceitação unânime. A primeira tentativa de uma definição formal para a interpretabilidade foi feita por (TARSKI; MOSTOWSKI; ROBINSON, 1953), que formulou uma definição matemática dentro do contexto da lógica clássica, estabelecendo-se a base para identificar teorias interpretáveis. Para Tarski, Mostowski e Robinson (1953), se considerar T e S como teorias formais, T será interpretável em S se e somente se houver uma forma de passar de T para S , certificando-se de que cada teorema de T pode ser traduzido e provado em S .

Dentro do contexto dos sistemas *fuzzy*, pode-se encontrar na literatura (MENCAR; CASTELLANO; FANELLI, 2005) uma definição de interpretabilidade semelhante à que foi proposta por (TARSKI; MOSTOWSKI; ROBINSON, 1953), em que é feita uma distinção entre uma linguagem formal L (lógica *fuzzy*), utilizada para descrever o modelo em análise, e uma linguagem orientada ao usuário L' (geralmente a língua natural), utilizada para explicar o modelo para o usuário. Se o sistema é interpretável, então a tradução de L para L' deve ser feita pelo usuário com um esforço pequeno. De uma maneira informal, as pessoas dizem que um modelo é interpretável se elas são capazes de descrever e explicar este modelo facilmente.

Uma definição bastante aceita foi dada por Bodenhofer e Bauer (2003): “interpretabilidade

é a possibilidade de estimar o comportamento do sistema por meio da leitura e compreensão da base de regras”.

2.6.1 Taxonomia para Índices de Interpretabilidade de SFBR

Conforme comentado anteriormente, a legibilidade de um SFBR influencia diretamente a compreensão do sistema. Por sua vez, em uma visão global, os dois fatores que influenciam a legibilidade são: o conjunto de regras e a partição *fuzzy* (GUILLAUME; CHARNOMORDIC, 2004) (Espinosa & Vandewalle, 2000; Guillaume & Charnomordic, 2004).

Além disso, como explicado por (MENCAR; FANELLI, 2008), é possível distinguir dois níveis de interpretabilidade principais: baixo nível (ou nível do conjunto *fuzzy*); e alto nível (ou nível da regra *fuzzy*).

Baseados nestes fatores e nas restrições inerentes à legibilidade e compreensão de um SFBR, Zhou e Gan (2008) apresentaram uma taxonomia de interpretabilidade em SFBR (ZHOU; GAN, 2008). Mais tarde, essa taxonomia foi estendida por Alonso, Magdalena e González-Rodríguez (2009) fundamentados em um estudo completo sobre as restrições em interpretabilidade mais utilizada na modelagem *fuzzy* (MENCAR; FANELLI, 2008). Essa taxonomia é ilustrada na Figura 2.1.

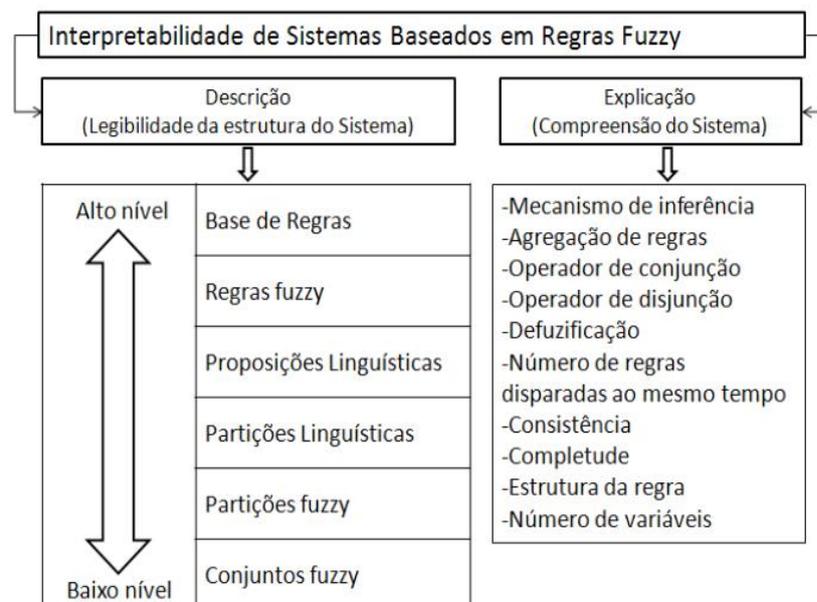


Figura 2.1: Caracterização da interpretabilidade. Figura baseada em (ALONSO; MAGDALENA; GONZÁLEZ-RODRÍGUEZ, 2009).

A descrição (legibilidade da estrutura) de um SFBR pode ser analisada observando diferentes níveis de abstração, como ilustrado na parte esquerda da Figura 2.1. O nível mais baixo

corresponde ao nível dos conjuntos *fuzzy*. Ele inclui as restrições exigidas para construir conjuntos *fuzzy* interpretáveis levando em conta propriedades matemáticas (convexidade, protótipos, etc.) das funções de pertinência. No segundo nível, existem algumas restrições com relação à combinação de vários conjuntos *fuzzy* para formar uma partição *fuzzy*.

O uso de variáveis linguísticas favorece a legibilidade, mas não é o suficiente para garantir a interpretabilidade. Assim, algumas restrições linguísticas devem ser aplicadas à definição da partição *fuzzy* para que elas sejam interpretáveis. Com o intuito de satisfazer algumas restrições semânticas (distinguidade, cobertura, normalidade, convexidade, etc.), Ruspini definiu um tipo especial de partição chamada *partição fuzzy forte* (Strong Fuzzy Partition), em que as funções de pertinência se sobrepõem duas a duas no valor de 50% (RUSPINI, 1969). Podemos ver na Figura 2.2 um exemplo de partição *fuzzy forte* com cinco conjuntos *fuzzy* se sobrepondo exatamente no grau de pertinência de 50%.

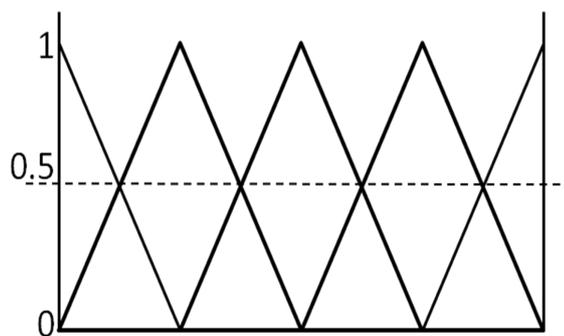


Figura 2.2: Partição *fuzzy forte*

Devido às limitações da memória humana, torna-se essencial trabalhar com partições *fuzzy fortes* compostas por um número pequeno de termos linguísticos. De acordo com psicólogos (MILLER, 1956; SAATY; OZDEMIR, 2003), o limite da capacidade humana de processamento de informação é 7 ± 2 termos linguísticos.

O uso de partições *fuzzy fortes* resulta em partições *fuzzy* interpretáveis, pois estas mantêm estruturas claras e transparentes. No entanto, do ponto de vista da interpretabilidade, outra questão importante é a escolha correta dos rótulos linguísticos, que deve ser feita de acordo com o contexto do problema. Este é o terceiro nível de abstração da Figura 2.1.

Uma vez definidos os termos linguísticos, juntamente com a sua semântica, eles podem ser usados para expressar proposições linguísticas (quarto nível de abstração). As restrições relacionadas às proposições linguísticas incluem o uso de modificadores linguísticos e a composição de termos linguísticos. Deste modo, várias proposições são combinadas para formar regras *fuzzy* que descrevem o comportamento do sistema (quinto nível de abstração). As restri-

ções relacionadas às regras *fuzzy* consideram o tamanho do consequente da regra e o número de variáveis em cada regra.

Além da análise de cada regra individual, há a necessidade de estudar a combinação de várias regras, alcançando assim o maior nível de abstração. Neste nível de abstração são considerados o tamanho da BR (número total de regras na BR) e o tamanho da regra (número total de premissas).

Quando todas as restrições listadas na parte esquerda da Figura 2.1 são satisfeitas, a interpretabilidade do SFBR é garantida do ponto de vista estrutural. Na prática, satisfazer todas essas restrições é quase impossível, pois elas representam um grande conjunto de condições que normalmente direcionam o sistema para um baixo nível de acurácia. De fato, Fazendeiro, Valente de Oliveira e Pedrycz (2007) mostraram que, quando se desconsidera a propriedade de partições *fuzzy* fortes, pode-se obter sistemas mais precisos, mas à custa de diminuir a legibilidade.

A explicação (compreensão) do comportamento de um SFBR, lado direito da Figura 2.1, leva em consideração a interpretação adequada da própria regra e como as regras são combinadas.

A explicação do comportamento do sistema também leva em consideração o sistema de inferência. Há diversos tipos de regras *fuzzy* (regra gradual, regra possibilística, regras com pesos, etc.) que podem ser utilizadas, sendo que cada uma delas possui um mecanismo de inferência específico (Dubois & Prade, 1996). Desta forma, a escolha da regra apropriada é um aspecto importante durante a modelagem do sistema, não apenas pela acurácia, mas também pela semântica das regras *fuzzy*. Cada regra *fuzzy* tem uma semântica diferente que corresponde a uma interpretação diferente.

O nível de inferência também inclui as definições do operador *fuzzy* para conjunção, disjunção, agregação e defuzificação. Toda a BR deve ser consistente (não pode existir redundância, contradições, etc.) e deve abranger as situações mais prováveis.

De acordo com Guillaume (2001), completude significa que para qualquer vetor de entrada possível, pelo menos uma regra é disparada. A completude da BR depende do conjunto de dados disponíveis. Quanto maior o número de casos no conjunto de dados, maior será o número de casos cobertos pela BR. Entretanto, a coleta de dados implica fazer muitos experimentos que custam tempo. Além disso, alguns experimentos não são viáveis porque exigem condições extremas. Sendo assim, os requisitos de completude variam dependendo da aplicação.

A falta de completude é normalmente tratada usando regras *default* que só são disparadas quando nenhuma outra regra foi disparada. O objetivo é evitar situações anormais que possam

produzir algum dano. Para alguns sistemas, a completude da BR não é essencial, pois envolvem interação com os seres humanos. Como exemplo destes sistemas tem-se sistemas de apoio à decisão, classificação supervisionada, diagnósticos, etc.

Levando em consideração que como o resultado de um mecanismo de inferência *fuzzy* várias regras podem ser disparadas ao mesmo tempo para um dado vetor de entrada, a compreensão do sistema está diretamente ligada ao número de regras que podem ser disparadas simultaneamente. Quanto menor o número de regras disparadas ao mesmo tempo, maior será a compreensão que teremos do sistema. Um modelo composto por milhares de regras, onde no máximo dez regras são disparadas simultaneamente, pode ser visto como mais compreensível do que um modelo incluindo apenas cem regras, onde a maioria delas é disparada simultaneamente (ALONSO; MAGDALENA; GONZÁLEZ-RODRÍGUEZ, 2009).

Apenas a leitura da BR não é suficiente para entender o comportamento do sistema, pois a saída é obtida como resultado da combinação de um conjunto de regras. Por isso, explicar a saída de um SFBR não é uma tarefa simples, já que ainda há uma lacuna entre a descrição do sistema e a compreensão do seu comportamento.

Outra taxonomia encontrada na literatura foi proposta por (GACTO; ALCALÁ; HERRERA, 2011). Esta taxonomia é baseada na complexidade e na semântica da BR e das partições *fuzzy*, conforme segue:

- Quadrante 1 (Q_1): complexidade em nível da BR;
- Quadrante 2 (Q_2): complexidade em nível das partições *fuzzy*;
- Quadrante 3 (Q_3): semântica em nível da BR;
- Quadrante 4 (Q_4): semântica em nível das partições *fuzzy*.

No quadrante Q_1 estão inseridos os trabalhos que tentam reduzir ou controlar a complexidade da BR. As medidas mais utilizadas por trabalhos que pertencem a Q_1 são:

- **Número de regras:** O conjunto de regras *fuzzy* deve ser o menor possível sob condições nas quais o desempenho do modelo seja mantido em um nível satisfatório.
- **Número de condições:** O número de condições no antecedente de uma regra não deve exceder o limite de 7 ± 2 condições distintas. Além disso, o número de condições deve ser o menor possível, a fim de facilitar a leitura das regras. O desempenho do modelo também deve ser mantido em um nível satisfatório.

Para o quadrante Q_2 , as medidas mais utilizadas para controlar a complexidade em nível das partições *fuzzy* são:

- **Número de atributos ou variáveis:** Esta medida é utilizada principalmente para reduzir a dimensionalidade em problemas de alta dimensionalidade. A redução do número de atributos pode melhorar a legibilidade da BC.
- **Número de funções de pertinência:** Para controlar a complexidade em nível das partições *fuzzy*, é necessário que haja um número moderado de funções de pertinência. Em geral considera-se que o número de funções de pertinência não deve exceder o limite de 7 ± 2 funções distintas. Quanto maior for o número de funções de pertinência, mais a acurácia do sistema pode aumentar, porém a relevância das funções de pertinência pode diminuir.

O quadrante Q_3 está relacionado com as medidas ou propriedades que controlam a interpretabilidade semântica em nível da BR. Este quadrante leva em conta principalmente as seguintes propriedades:

- **Consistência da BR:** observa a ausência de regras contraditórias na BR, ou seja, regras com antecedentes iguais devem ter consequentes iguais.
- **Número de regras disparadas ao mesmo tempo:** consiste em minimizar o número de regras de disparo que são ativadas para um determinado vetor de entrada.

Durante o processo de geração de SFBR mais precisos, pode-se alterar algumas propriedades dos conjuntos *fuzzy* e, conseqüentemente, afetar a sua interpretabilidade. A Figura 2.3 mostra um exemplo de uma partição *fuzzy* com enorme sobreposição entre as funções de pertinência, o que deteriora a interpretabilidade global do ponto de vista da interpretabilidade semântica.

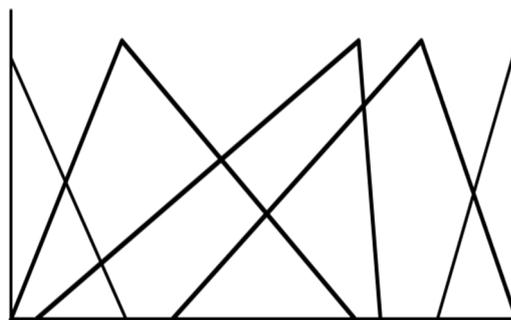


Figura 2.3: Particionamento com baixo significado semântico

No quadrante Q_4 há muita preocupação com a manutenção da interpretabilidade semântica em nível da partição *fuzzy*. Este problema tem sido abordado através da aplicação de algumas restrições para a definição das funções de pertinência a fim de preservar ou melhorar algumas propriedades desejáveis. Algumas das propriedades mais importantes que devem ser lembradas ao definir a partição *fuzzy* são (ISHIBUCHI; YAMAMOTO, 2004a), são (Ishibuchi & Yamamoto, 2004):

- **Cobertura:** As funções de pertinência devem cobrir o universo de discurso inteiro da variável, e cada ponto do domínio deve ter pertinência diferente de zero a pelo menos um dos conjuntos *fuzzy*;
- **Normalização:** Uma função de pertinência é normal se existe pelo menos um ponto do domínio em que o grau de pertinência deste ponto na função de pertinência seja igual a 1.
- **Distinguiabilidade:** As funções de pertinência devem representar um termo linguístico com um significado claro e devem ser facilmente distinguíveis entre si.
- **Complementaridade:** Para cada elemento do domínio, a soma de todos os graus de pertinência deve estar próximo a 1. Isto garante a distribuição uniforme entre os elementos.

Uma visão geral desta taxonomia pode ser vista na Tabela 2.1, em que são expostas as medidas consideradas em cada quadrante.

	Nível da Base de Regras	Nível das partições <i>fuzzy</i>
Interpretabilidade baseada em complexidade	Q_1	Q_2
	Número de regras	Número de funções de pertinência
	Número de condições	Número de atributos
Interpretabilidade baseada em semântica	Q_3	Q_4
	Consistência da regra	Cobertura
	Regras disparadas ao mesmo tempo	Normalização
	Transparência da estrutura da regra	Distinguiabilidade
	Igualdade	Complementaridade
		Medidas relativas

Tabela 2.1: Taxonomia para a interpretabilidade. Traduzida de (GACTO; ALCALÁ; HERRERA, 2011)

2.7 Considerações finais

Neste capítulo foram apresentados os principais conceitos sobre conjuntos *fuzzy*, em especial os Sistemas de Classificação *Fuzzy*, alvo de estudo deste trabalho.

Também foi apresentado o conceito de interpretabilidade em SFBR e alguns trabalhos relacionados, mostrando os fatores que envolvem a interpretabilidade de um Sistema *Fuzzy* Baseado em Regras.

No próximo capítulo serão apresentados os conceitos básicos de Algoritmos Genéticos, Algoritmos Genéticos Multiobjetivos e Sistemas *Fuzzy* Genéticos.

Capítulo 3

COMPUTAÇÃO EVOLUTIVA

3.1 Considerações Iniciais

A Computação Evolutiva (CE), também conhecida como Computação Evolucionária, é um ramo de pesquisa que compreende um conjunto de técnicas de busca e otimização baseadas nos princípios da evolução biológica, como a seleção natural e a herança genética, para resolver diferentes tipos de problemas. Essas técnicas estão sendo cada vez mais utilizadas para resolver uma grande variedade de problemas, que vão desde as aplicações práticas na indústria e comércio até a investigação científica (EIBEN et al., 2007).

Por meio de técnicas da CE, cria-se uma população de indivíduos que se reproduzem e competem entre si pela sobrevivência. Os melhores indivíduos possuem maiores chances de sobreviver e de transferir suas características às novas gerações.

As técnicas principais da CE são as Estratégias Evolutivas (EE), a Programação Evolutiva (PE), a Programação Genética (PG) e os Algoritmos Genéticos (AG) (PEDRYCZ; GOMIDE, 1998; BANZHAF et al., 1998). Todas estas técnicas utilizam o mesmo princípio básico da CE, mas operam de formas distintas.

As EE codificam uma população em um vetor único de indivíduos com valores reais e a sua ideia principal é a combinação dos indivíduos, mediante cruzamento, para gerar um descendente que substituirá o pior indivíduo da população. O primeiro algoritmo usando uma estratégia evolutiva foi desenvolvido em 1964 na Universidade Técnica de Berlim (KUSIAK, 2000).

Já a PE utiliza a predição do comportamento de máquinas de estado finito e sua otimização (PEDRYCZ; GOMIDE, 1998). Dessa forma, cada indivíduo representa uma máquina de estado finito. A seleção dos indivíduos é baseada no elitismo.

A PG é uma técnica de geração automática de programas de computador proposta por Koza (1992), inspirada na teoria dos AG proposta por Holland (1975). É possível criar e manipular software geneticamente usando-se PG e aplicando conceitos herdados da biologia para gerar programas de computador automaticamente.

Os AG são algoritmos que modelam uma solução para um problema específico em uma estrutura de dados e aplicam operadores que recombina essas soluções preservando informações críticas. Os AG foram concebidos por Holland (1975), com o objetivo inicial de estudar os fenômenos relacionados à adaptação das espécies e da seleção natural que ocorrem na natureza, bem como desenvolver uma maneira de incorporar estes conceitos aos computadores (MITCHELL, 1998). Os AG e a PG são as duas principais frentes de pesquisa na computação evolutiva. Na próxima seção são apresentados os conceitos principais de AG. A PE, PG e as EE não serão estudadas nesta proposta. Informações mais detalhadas sobre estas técnicas podem ser encontradas em diversos trabalhos (BÄCK; SCHWEFEL, 1993; De Jong, 2006; EIBEN et al., 2007).

3.2 Algoritmos Genéticos

Os AG podem ser definidos como métodos iterativos que contêm uma população com um número fixo de indivíduos, em que cada um deles é representado por uma sequência linear finita de símbolos, denominada cromossomo, que codificam uma possível solução para um dado problema, ou, dependendo da codificação, parte da solução. Essas possíveis soluções são codificadas em um espaço, chamado de espaço de busca, que pode conter todas as soluções possíveis para o problema em questão. A cada cromossomo é associado um valor de aptidão e este valor representa quão adequada ao problema é a solução que ele codifica. Dessa forma, os indivíduos podem ser alterados até que se obtenha um indivíduo que codifique uma solução satisfatória para o problema abordado.

Os AG são frequentemente usados em problemas de busca, tendo encontrado ampla aplicação em diversas áreas científicas, entre as quais podem ser citadas aquelas relativas a problemas de otimização de soluções, aprendizado de máquina, desenvolvimento de estratégias e fórmulas matemáticas, análise de modelos econômicos, problemas de engenharia, diversas aplicações na biologia tais como simulação de bactérias, sistemas imunológicos, ecossistemas e descoberta de topologias e propriedades de moléculas orgânicas (GOLDBERG, 1989; HAUPT; HAUPT, 2004; CHAMBERS, 2001; MICHALEWICZ, 1996; MITCHELL, 1998).

O funcionamento de um AG dá-se a partir do processo de inicialização, que cria uma *po-*

pulção de indivíduos representando soluções iniciais que são, então, submetidas ao processo de evolução, que passa pelas seguintes etapas: avaliação, seleção, cruzamento e mutação. Cada uma destas etapas serão abordadas a seguir.

3.2.1 Avaliação

É nesta fase que é dado o primeiro passo para a seleção. Cada indivíduo recebe uma avaliação de acordo com o seu grau de aptidão, ou seja, mede-se quão bom é o indivíduo para solucionar o problema em questão. Levando-se em conta que esta tarefa é executada para cada indivíduo a cada geração, pode-se dizer que o seu custo computacional é relativamente alto.

3.2.2 Seleção

Na etapa de seleção, os indivíduos são selecionados para a próxima etapa, o cruzamento. Utiliza-se o grau de aptidão de cada indivíduo para fazer um sorteio no qual os indivíduos mais aptos possuem maior probabilidade de serem escolhidos.

A seguir são listados alguns métodos de seleção.

Roleta: Neste método os indivíduos de uma população são selecionados para a próxima geração utilizando uma roleta. Cada indivíduo da população é representado na roleta por uma fatia proporcional ao seu índice de aptidão. Assim, indivíduos com maior aptidão ocupam fatias maiores da roleta, enquanto indivíduos com menor aptidão ocupam fatias menores. Para a seleção dos indivíduos, a roleta é girada N vezes, onde N é número da população inicial. A cada vez que a roleta parar de girar, o cromossomo selecionado pelo marcador será copiado para a próxima geração. Cromossomos com maior espaço na roleta terão maior chance de serem selecionados.

Na Figura 3.1 é apresentada uma roleta que reflete os valores de aptidão dos 5 indivíduos (S1, S2, S3, S4 e S5) de uma população, sendo que o indivíduo S2 (com a maior fatia) foi selecionado pelo marcador.

Torneio: Quando este método é utilizado, um número n de indivíduos da população é escolhido de modo aleatório e com mesma probabilidade. O cromossomo com maior aptidão dentre estes n cromossomos é selecionado para a população intermediária. Este processo se repetirá até que a população intermediária seja preenchida.

Pode-se, ainda, adicionar a esses métodos uma técnica chamada *elitismo*. Nesta técnica todos os indivíduos são ordenados de acordo com o seu valor de aptidão e os indivíduos que



Figura 3.1: Método de seleção por roleta.

ficarem abaixo do valor médio são descartados. Esta técnica resulta em uma população que terá indivíduos com valor de aptidão acima da média da população anterior. Esta técnica também garante que o melhor indivíduo obtido na geração atual estará presente na próxima geração que está sendo gerada, preservando suas características genéticas.

3.2.3 Cruzamento

Selecionados os indivíduos, eles passam com uma probabilidade pré-estabelecida pelo processo de cruzamento. Essa probabilidade é chamada de taxa de cruzamento. Neste processo são selecionados dois indivíduos para o cruzamento, gerando dois novos indivíduos que irão compor a próxima geração substituindo, assim, os dois indivíduos selecionados.

Dentre os métodos de cruzamento pode-se destacar (GOLDBERG, 1989):

Cruzamento de 1 ponto Neste método, é escolhido um ponto de corte p qualquer onde $1 < p < w$, sendo w o comprimento do cromossomo. O primeiro cromossomo filho receberá os mesmos genes do pai 1 compreendidos entre o índice 1 e p . Os genes seguintes serão preenchidos com os genes do pai 2 compreendidos entre o índice $p + 1$ e w . O segundo cromossomo filho terá o preenchimento pela forma contrária, ou seja, receberá os mesmos genes do pai 2 compreendidos entre o índice 1 e p . Os genes seguintes serão preenchidos com os genes do pai compreendidos entre o índice $p + 1$ e w . A Figura 3.2 ilustra um exemplo deste tipo de cruzamento.

Cruzamento de 2 pontos No cruzamento de 2 pontos são escolhidos aleatoriamente 2 pontos, p_1 e p_2 , dos cromossomos pais tal que $p_1 < p_2$. O primeiro cromossomo filho terá os mesmos genes do cromossomo do pai 1. Entretanto, a região que compreende p_1 e p_2 terá os genes do pai 2. O segundo cromossomo filho terá os mesmos genes do cromossomo do pai 2. Entretanto, a região que compreende p_1 e p_2 terá os genes do pai 1. A Figura 3.3 ilustra um

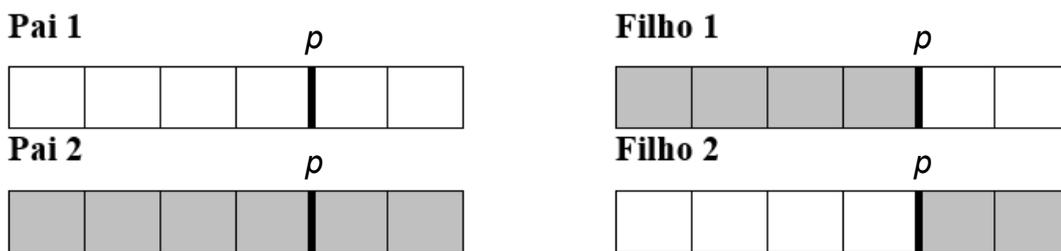


Figura 3.2: Cruzamento de 1 ponto.

exemplo deste tipo de cruzamento.

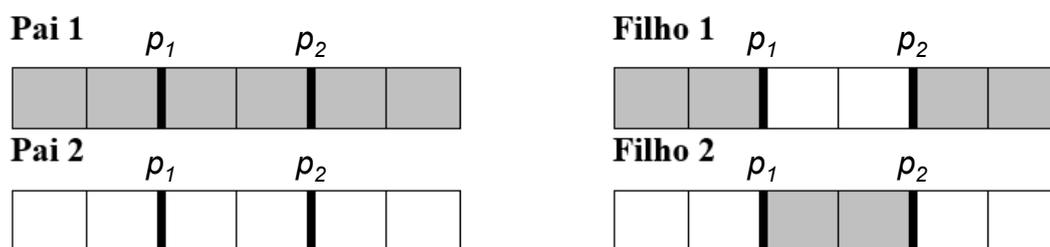


Figura 3.3: Cruzamento de 2 pontos.

3.2.4 Cruzamento Aritmético

O cruzamento aritmético é definido como uma combinação linear de dois vetores. Se \mathbf{x}_1 e \mathbf{x}_2 são cruzados, os descendentes serão $\mathbf{x}'_1 = a \cdot \mathbf{x}'_1 + (1 - a) \cdot \mathbf{x}'_2$ e $\mathbf{x}'_2 = a \cdot \mathbf{x}'_2 + (1 - a) \cdot \mathbf{x}'_1$, onde a é um número aleatório no intervalo $[0, 1]$.

3.2.5 Mutação

A mutação é um processo que garante a exploração de diversas alternativas e é aplicada aleatoriamente sobre os indivíduos da geração atual com uma taxa de probabilidade pré-definida chamada *taxa de mutação*. A mutação nada mais é do que algum tipo de mudança que deve ser realizada sobre um gene. Essa mudança depende de como o gene foi projetado. Em um cromossomo com codificação binária, por exemplo, a mutação altera um gene com valor 1 para valor 0 e vice-versa. Em um cromossomo com codificação real ou inteira, o gene é alterado trocando o valor atual do gene por algum outro valor dentre os possíveis valores válidos para o gene em questão.

Apesar de a aleatoriedade ser uma característica intrínseca aos AG, eles fazem uma busca direcionada. Os novos indivíduos gerados possuem características da população anterior, e

essas características são usadas como histórico que direciona a busca para um resultado melhor que o resultado gerado pela população anterior.

No Algoritmo 3.1 são apresentados os passos básicos de um AG em que o critério de parada é dado pelo parâmetro *maxGeracoes*.

Algoritmo 3.1: AlgoritmoGenetico

```

1 Início
2    $t := 1;$ 
3   inicializa( $P_t$ ); //  $P_t$  é a população da geração  $t$ 
4   Enquanto  $t \leq \text{maxGeracoes}$  Faça
5     avalia( $P_t$ );
6      $t := t + 1;$ 
7     seleciona( $t, P_{t-1}$ );
8     aplicaCruzamento( $P_t$ );
9     aplicaMutacao( $P_t$ );
10  Fim-Enquanto
11 Fim

```

3.3 Otimização Multiobjetivo

Em um problema de otimização simples (mono-objetivo) o espaço de busca geralmente é bem definido. Mas quando o problema tem múltiplos objetivos, contraditórios entre si, não existe apenas uma única solução ótima, mas sim um conjunto de possíveis soluções.

Mesmo que alguns problemas do mundo real possam ser reduzidos a um único objetivo, muitas vezes é difícil resumir todas as características do problema em um único objetivo e resolvê-lo utilizando técnicas de otimização mono-objetivo, como, por exemplo, os AG (ABRAHAM; JAIN; GOLDBERG, 2005). Assim, definir múltiplos objetivos frequentemente é a melhor forma de resolver um problema multiobjetivo.

A otimização multiobjetivo busca encontrar soluções de problemas que possuem objetivos que sejam conflitantes, ou seja, caso seja possível melhorar o resultado de um dos objetivos, o outro objetivo do problema ficará prejudicado. Diz-se que uma função f_1 é conflitante com uma função f_2 quando não é possível, ao mesmo tempo, melhorar o valor de ambas.

Um exemplo didático que ilustra bem este tipo de problema é o de uma viagem de automóvel em que se deseja minimizar o tempo de viagem e o consumo de combustível (custo). Sabe-se que quanto mais se acelera um veículo para diminuir o tempo de viagem, mais se gasta combustível. Matematicamente, o problema de otimização multiobjetivo pode ser formulado

como apresentado por (DIAS; VASCONCELOS, 2002), pela Equação 3.1, como segue.

$$\left. \begin{array}{ll} \text{maximizar/ minimizar} & f_m(\mathbf{x}), \quad m = 1, 2, \dots, M \\ \text{restrita a} & g_j(\mathbf{x}) \geq 0, \quad j = 1, 2, \dots, J \\ & h_k(\mathbf{x}) = 0, \quad k = 1, 2, \dots, K \\ & \mathbf{x}_i^{(L)} \leq \mathbf{x}_i \leq \mathbf{x}_i^{(U)}, \quad i = 1, 2, \dots, n \end{array} \right\} \quad (3.1)$$

em que \mathbf{x} é o vetor de n variáveis de decisão $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$. Os valores $x_i^{(L)}$ e $x_i^{(U)}$, representam o mínimo e máximo valores respectivamente para a variável x_i . Estes limites definem o *espaço das variáveis de decisão* ou *espaço de decisão* D . Além disso, o vetor \mathbf{x} será referido também como *solução*.

As J desigualdades (g_j) e as K igualdades (h_k) são chamadas de funções de restrição. Uma solução \mathbf{x} factível será aquela que satisfaça as $J + K$ funções de restrição e os $2n$ limites. Caso contrário a solução será não factível. O conjunto de todas as soluções factíveis formam a *região factível* ou *espaço de busca* S .

Cada uma das M funções objetivo $f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})$ pode ser maximizada ou minimizada. O vetor funções objetivo $\mathbf{f}(\mathbf{x})$ forma um espaço multi-dimensional chamado espaço de objetivos Z . Então, para cada solução \mathbf{x} no espaço de decisão, existe um $\mathbf{f}(\mathbf{x})$ no espaço de objetivos. Esta é uma diferença fundamental em relação à otimização de objetivos simples, cujo espaço de objetivos é unidimensional.

O mapeamento acontece então entre um vetor \mathbf{x} (n -dimensional) e um vetor $\mathbf{f}(\mathbf{x})$ (M -dimensional). Por exemplo, se cada elemento de \mathbf{x} e $\mathbf{f}(\mathbf{x})$ é um número real, então $\mathbf{f}(\mathbf{x})$ é definida como $\mathbf{f}(\mathbf{x}): \mathbb{R}^n \rightarrow \mathbb{R}^M$.

3.3.1 Soluções Pareto-ótimas

Tomar decisões implica em um processo que consiste de vários fatores e cujo objetivo é encontrar a melhor solução. Em alguns casos, é possível encontrar várias soluções boas, sendo que nenhuma é quantitativamente melhor do que a outra. Por exemplo, ao se comprar um carro, pode-se supor-se que se está procurando um carro com o menor preço e o maior conforto. A Figura 3.4 ilustra várias alternativas de escolha.

O objetivo é minimizar o custo e maximizar o conforto. Neste caso tem-se cinco possíveis opções de compra. Intuitivamente, descarta-se a solução 1, já que a solução 5 fornece mais conforto por igual preço. A solução 2 também é descartada, já que as soluções 3 e 5 fornecem

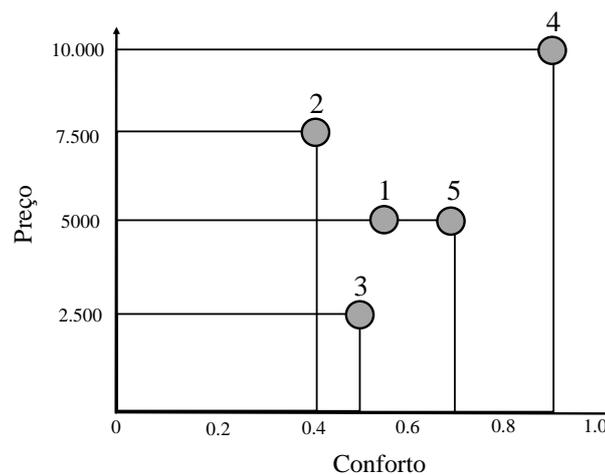


Figura 3.4: Exemplo de compra de carro, considerando o custo e o conforto.

mais conforto que a solução 2 e por um preço menor. Tem-se então três soluções (3, 4 e 5) que são boas alternativas de compra. Em termos quantitativos, nenhuma é melhor do que a outra, pois uma é mais confortável, porém menos barata, e vice-versa. Existe então um balanceamento entre os objetivos. Quanto maior o conforto, maior é o preço do carro e quanto mais barato, menor é o conforto do carro.

Diz-se que uma solução *domina* uma outra se os seus valores são melhores em todos os objetivos, ou é melhor em um dos objetivos e empata nos outros. Por exemplo, a solução 1 domina a solução 2, mas é dominada pelas soluções 4 e 5. Nenhuma solução domina as soluções 3, 4, e 5. Pode-se dizer que as soluções 3, 4, e 5 são igualmente boas. Portanto, existe um conjunto de soluções ótimas. A este conjunto dá-se o nome de *conjunto de soluções não dominadas*. As outras soluções (1 e 2) formam o *conjunto de soluções dominadas*.

Considerando que os pontos não dominados estão em um espaço contínuo, pode-se desenhar uma curva ligando esses pontos. Todos os pontos contidos na curva formam a *frente de Pareto* ou *fronteira de Pareto*.

3.3.2 Metas em Problemas de Otimização Multiobjetivo

Em um problema de otimização multiobjetivo, todas as soluções Pareto-ótimas, ou seja, as soluções encontradas na fronteira de Pareto, são igualmente importantes. Deb et al. (2002a) destaca duas importantes metas em otimização multiobjetivo:

- Encontrar um conjunto de soluções que estejam o mais próximo possível da fronteira de Pareto;

- Encontrar um conjunto de soluções não dominadas com a maior diversidade possível.

A primeira meta é comum para qualquer processo de otimização. Soluções muito distantes da Fronteira de Pareto não são desejáveis, pois isto indica que estas soluções estão distantes das melhores soluções. Porém, encontrar a maior diversidade dentro das soluções não dominadas é uma meta específica para a otimização multiobjetivo. A Figura 3.5a apresenta uma fronteira de Pareto com uma boa distribuição de soluções quando comparada com a da apresentada na Figura 3.5b.

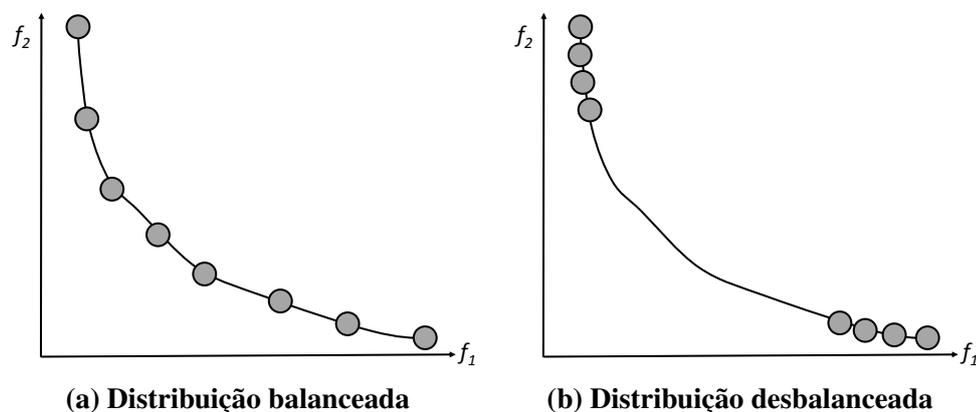


Figura 3.5: Distribuição das soluções ao longo da fronteira de Pareto

Na Figura 3.5b as soluções estão distribuídas apenas em algumas regiões, sendo que cada região privilegia um dos objetivos, o que caracteriza essas soluções como *desbalanceadas*. É necessário assegurar a maior cobertura possível da fronteira, fazendo com que se obtenha um conjunto de soluções com os objetivos balanceados.

3.3.3 Diferenças entre Otimização Mono-objetivo e Multiobjetivo

Deb et al. (2002a) identifica três diferenças importantes entre a otimização multiobjetivo e a otimização de mono-objetivo:

- Em problemas de otimização mono-objetivo, a meta é achar uma solução ótima global (máximo ou mínimo). Em problemas multiobjetivo pode existir mais de um ótimo global. Achar o conjunto de soluções da fronteira de Pareto é tão importante quanto preservar a diversidade neste conjunto. Um algoritmo eficiente para otimização multiobjetivo deve considerar ambos os aspectos;
- Em problemas de otimização multiobjetivo são considerados dois espaços (o de variáveis e o de objetivos) ao invés de um. Problemas de otimização mono-objetivo trabalham

unicamente no espaço de variáveis, já que procuram apenas uma solução no espaço de objetivos. Manter a diversidade em ambos espaços aumenta a complexidade do problema, dado que a proximidade entre duas soluções no espaço de variáveis não implica proximidade no espaço de objetivos;

- Os métodos tradicionais de otimização multiobjetivo estão baseados em uma função simples que pondera cada objetivo. Podem também tratar cada objetivo separadamente, utilizando os demais objetivos como restrições. Portanto, um problema de otimização multiobjetivo pode ser convertido em um problema de otimização simples.

3.3.4 Operador de Dominância de Pareto

Os problemas multiobjetivos, diferentemente dos problemas mono-objetivos, possuem mais de uma solução. Estas soluções são conhecidas como *soluções não-dominadas* ou *soluções eficientes*.

Se existem M funções objetivo f_m , $m = 1, \dots, M$ o operador \triangleleft entre duas soluções, $x \triangleleft y$, significa que a solução x é melhor do que a solução y em pelo menos um objetivo em particular. Reciprocamente $x \triangleright y$ denota que a solução x é pior do que a solução y para algum objetivo.

Diz-se que uma solução $\mathbf{x}^{(1)}$ domina uma outra solução $\mathbf{x}^{(2)}$ (representado matematicamente como $\mathbf{x}^{(1)} \prec \mathbf{x}^{(2)}$) se as seguintes condições são satisfeitas:

- A solução $\mathbf{x}^{(1)}$ não é pior que $\mathbf{x}^{(2)}$ em todos os objetivos, ou seja, $f_m(\mathbf{x}^{(1)}) \not\triangleright f_m(\mathbf{x}^{(2)})$ para todo $m = 1, 2, \dots, M$;
- A solução $\mathbf{x}^{(1)}$ é estritamente melhor que $\mathbf{x}^{(2)}$ pelo menos em um objetivo, ou seja, $f_m(\mathbf{x}^{(1)}) \triangleleft f_m(\mathbf{x}^{(2)})$ pelo menos em um $m = 1, 2, \dots, M$.

Se ambas as condições são satisfeitas, pode-se dizer que $\mathbf{x}^{(1)}$ domina $\mathbf{x}^{(2)}$:

- $\mathbf{x}^{(1)}$ domina $\mathbf{x}^{(2)}$;
- $\mathbf{x}^{(1)}$ é não dominada por $\mathbf{x}^{(2)}$;
- $\mathbf{x}^{(1)}$ é não inferior que $\mathbf{x}^{(2)}$.

Se a solução $\mathbf{x}^{(1)}$ não é dominada por nenhuma solução $\mathbf{x}^{(i)}$ qualquer, em todo o espaço factível, diz-se que $\mathbf{x}^{(1)}$ é uma solução eficiente, não-dominada ou solução Pareto-ótima.

Na Figura 3.4, a solução 5 domina a solução 1 ($5 \prec 1$), e a solução 3 domina a solução 2 ($3 \prec 2$). Portanto, o conceito de dominância permite comparar soluções com múltiplos objetivos.

Assim, utilizando estas definições, quando um conjunto de soluções finito é encontrado, torna-se possível realizar comparações das soluções duas a duas, dividindo-as em um grupo chamado de soluções dominadas P e de soluções não-dominadas P' . As soluções de P' são não-dominadas por qualquer outra solução presente em P . Se o conjunto não-dominado P' abrange a totalidade do espaço de busca factível, ele é chamado de conjunto Pareto-ótimo global. A Figura 3.6 ilustra os espaços das variáveis de decisão e dos objetivos. É também mostrado nesta figura a fronteira Pareto-ótima global.

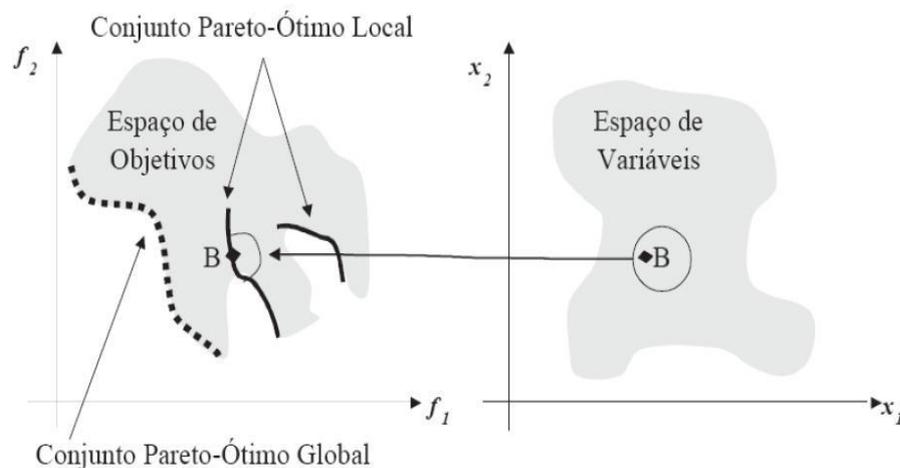


Figura 3.6: Soluções Pareto-ótimo locais e globais.

A fronteira Pareto-ótima ilustrada na Figura 3.6 é formada por valores das funções objetivo $\mathbf{f}(\mathbf{x}) = (f_1(x), \dots, f_m(x))$ correspondentes a cada solução no espaço de busca. Logo, para cada uma das soluções encontradas no espaço de variáveis, estas soluções são representadas no espaço dos objetivos, avaliando cada uma delas em cada um dos objetivos existentes.

3.4 Algoritmos Genéticos Multiobjetivo

Para resolver problemas de otimização multiobjetivo, pode-se empregar os AG, transformando o problema de otimização multiobjetivo em um problema de otimização mono-objetivo, utilizando algum critério de preferência para encontrar uma única solução. Essa solução deve possuir valores aceitáveis em todos os objetivos segundo o critério de preferências estabelecido, como, por exemplo, priorizar um ou mais objetivos dando pesos a eles. Se mais de uma solução for requerida, é necessário executar repetidamente o algoritmo modificando o critério de

preferências.

Por outro lado, pode-se também obter múltiplas soluções em uma execução do AG para resolver problemas multiobjetivo, tentando encontrar as soluções que estejam na fronteira de Pareto considerando a dominância destas soluções.

Os AG que consideram a dominância das soluções utilizando um conjunto de funções objetivo são denominados Algoritmos Genéticos Multiobjetivo (AGMO).

Goldberg (1989) afirma em seu livro que o uso de AGs para a solução de problemas multiobjetivo teve início quando Schaffer (1985) implementou a primeira versão de um AGMO denominado VEGA (*Vector Evaluated Genetic Algorithm*). No algoritmo proposto, considerando uma população de N indivíduos e K objetivos, divide-se esta população em K subpopulações com N/K indivíduos em cada uma delas. O operador de seleção dos AGs é aplicado separadamente para cada uma das subpopulações, isto é, para a subpopulação k considera-se apenas o k -ésimo objetivo para fins da seleção, e, posteriormente, unem-se estas subpopulações e aplica-se os outros operadores genéticos de cruzamento e mutação.

O trabalho de Coello (2006) apresenta uma visão geral da história da otimização multiobjetivo. Este trabalho divide os algoritmos existentes até então. A primeira delas possui algoritmos que possuem como característica a ênfase maior na simplicidade. Entre estes algoritmos destacam-se o VEGA, o *Nondominated Sorting Genetic Algorithm* (NSGA), o *Niched-Pareto Genetic Algorithm* (NPGA) e o *Multi-Objective Genetic Algorithm* (MOGA). A segunda geração dos algoritmos dá maior ênfase à eficiência. Entre os algoritmos classificados nesta geração estão: *Strength Pareto Evolutionary Algorithm* (SPEA), *Strength Pareto Evolutionary Algorithm II* (SPEA2), *Pareto Archived Evolution Strategy* (PAES) e o *Nondominated Sorting Genetic Algorithm II* (NSGA-II). A Tabela 3.1 mostra um resumo dos principais AGMO.

Dentre os diversos métodos citados para se encontrar soluções não-dominadas, neste trabalho o foco será dado ao algoritmo NSGA-II. O interesse neste algoritmo em especial se dá pelo fato de ele ser um dos mais populares e um dos que possuem melhores resultados reportados na literatura quando comparado a outras abordagens (ZITZLER; LAUMANN; THIELE, 2001; ZONG-YI et al., 2008; ANTONELLI; DUCANGE; MARCELLONI, 2014; MARTIN et al., 2014). O funcionamento do NSGA-II é detalhado a seguir.

3.4.1 *Nondominated Sorting Genetic Algorithm II (NSGA-II)*

Proposto por (DEB et al., 2002a), este método de otimização foi baseado em seu predecessor NSGA, que, por sua vez, foi implementado a partir de uma ideia de (GOLDBERG, 1989). A ideia

Tabela 3.1: Diferentes tipos de AGMO

Sigla	Nome do Algoritmo	Referência
VEGA	<i>Vector Evaluated Genetic Algorithm</i>	(SCHAFFER, 1985)
WBGA	<i>Weight Based Genetic Algorithm</i>	(HAJELA; LIN, 1992)
MOGA	<i>Multiple Objective Genetic Algorithm</i>	(FONSECA; FLEMING, 1993)
NSGA	<i>Non-Dominated Sorting Genetic Algorithm</i>	(SRINIVAS; DEB, 1994)
NPGA	<i>Niched-Pareto Genetic Algorithm</i>	(HORN; NAFPLIOTIS; GOLDBERG, 1994)
PPES	<i>Predator-Prey Evolution Strategy</i>	(LAUMANN; RUDOLPH; SCHWEFEL, 1998)
REMOEA	<i>Rudolph's Elitist Multi-Objective Evolutionary Algorithm</i>	(RUDOLPH, 2001)
NSGA-II	<i>Elitist Non-Dominated Sorting Genetic Algorithm</i>	(DEB et al., 2002a)
SPGA, SPEA2	<i>Strength Pareto Evolutionary Algorithm 1 e 2</i>	(ZITZLER; THIELE, 1998; ZITZLER; LAUMANN; THIELE, 2001)
TGA	<i>Thermodynamical Genetic Algorithm</i>	(KITA et al., 1996)
PAES	<i>Pareto-Archived Evolutionary Strategy</i>	(KNOWLES; CORNE, 1999)
MOMGA-I, MOMGA-II	<i>Multi-Objective Messy Genetic Algorithm</i>	(Van Veldhuizen, 1999)
PESA-I, PESA-II	<i>Pareto Envelope-Base Selection Algorithm</i>	(CORNE; KNOWLES; OATES, 2000; CORNE et al., 2001)

central do NSGA é classificar os indivíduos em fronteiras não-dominadas e aplicar um método para diversificar o máximo possível as soluções.

A diferença desta implementação em relação à de um AG mono-objetivo está apenas no modo com que o operador de seleção é empregado. Tanto os operadores de cruzamento quanto os operadores de mutação são idênticos aos utilizados na versão mono-objetivo do AG.

O NSGA-II apresentou soluções para problemas encontrados no NSGA, tais como a alta complexidade do procedimento proposto para a ordenação de não-dominância e a inexistência de elitismo. Para a solução dos problemas citados, o NSGA-II define um novo procedimento para a ordenação das soluções com base no critério de não-dominância e cria um novo conceito, chamado de *crowding distance* (distância de multidão), que se torna responsável por manter a diversidade da população. Também define um método chamado *crowded comparison* (comparação de multidão), que tem como objetivo comparar as soluções geradas.

O NSGA-II trabalha com uma população pai P para gerar uma população filha Q . Na primeira geração, a população inicial P_0 é ordenada por não-dominância. Cada uma das soluções

presentes na população P_0 recebe um valor de aptidão conforme o seu nível de não-dominância (1 para o melhor nível, 2 para o seguinte e assim por diante). Após este processo aplicam-se os operadores de seleção, cruzamento e mutação, obtendo-se assim uma população filha Q_0 . Ambas as populações existentes possuem tamanho N .

O pseudocódigo do algoritmo NSGA-II é apresentado no Algoritmo 3.2 com todos os detalhes do processo iterativo.

Algoritmo 3.2: NSGA-II

```

1 Início
2    $P$ : População pai;
3    $Q$ : População filha;
4    $T$ : Tamanho fixo para  $P$  e  $Q$ ;
5    $F_i$ : Conjunto de soluções na fronteira  $j$ ;
6    $n$ : Número da geração atual;
7    $N$ : Número máximo de gerações;
8   Gerar a população inicial  $P_0$  aleatoriamente;
9    $Q_0 = \{\}$ ;
10  Atribuir  $n = 0$ ;
11  Realizar a seleção, o cruzamento e a mutação em  $P_n$  para gerar a população filha  $Q_n$ ;
12  Fazer  $R_n = P_n \cup Q_n$ ;
13  Realizar a ordenação por não dominância em  $R_n$ , gerando  $F_i, i = \{1, \dots, v\}$  em  $R_n$ ;
14  Criar  $P_{n+1} = \{\}$ ;
15  Enquanto  $|P_{n+1}| + |F_i| \leq N$  Faça
16    | Copiar as soluções de  $F_i$  em  $P_{n+1}$ ;
17    |  $i = i + 1$ 
18  Fim-Enquanto
19  Calcular a distância de multidão para cada solução em  $F_i$ ;
20  Ordenar  $F_i$  decrescentemente de acordo com a distância de multidão de cada solução;
21  Copiar as primeiras  $N - |P_{n+1}|$  soluções ordenadas de  $F_i$  para  $P_{n+1}$ ;
22  Se  $n \geq N$  Então
23    | Pare;
24  Senão
25    | Atribuir  $n = n + 1$  e voltar ao passo 10;
26  Fim-Se
27 Fim

```

No NSGA-II é necessário calcular a distância de multidão para cada solução em uma fronteira F_i . A distância de multidão d de uma solução j , denotada por d_j , representa uma estimativa do perímetro formado pelo cubóide cujos vértices são os seus vizinhos mais próximos. A Figura 3.7 mostra a distância de multidão para uma solução i .

O cálculo da distância de multidão pode ser visto em detalhes no Algoritmo 3.3.

onde:

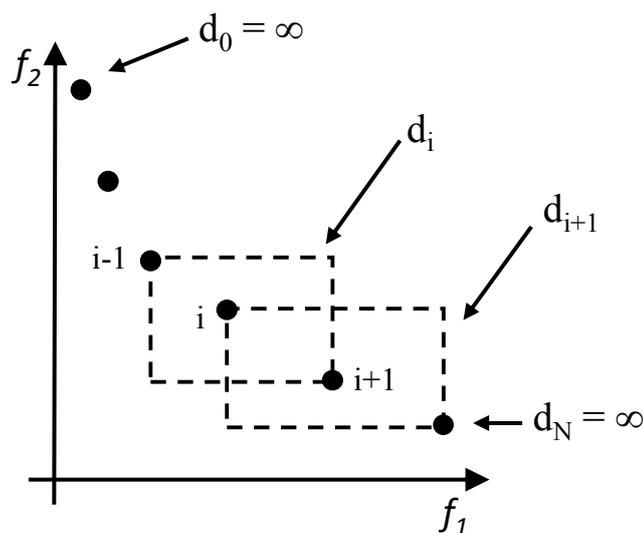


Figura 3.7: Cálculo da distância de multidão do NSGA-II em um espaço bidimensional (DEB et al., 2002a)

Algoritmo 3.3: Distância de Multidão

- 1 F_i : conjunto de soluções na fronteira i v_i : número de soluções na fronteira i m : número de funções objetivo
 - 2 Para cada solução j em F_i atribui-se $d_j = 0$ Para cada função objetivo $k = 1, 2, \dots, m$
 - 3 Gerar a lista de soluções I^m igual ao conjunto soluções de F_i ordenadas decrescentemente por f_k
 - 4 Fazer $d_{I_1^m} = d_{I_{v_i}^m} = \infty$
 - 5 Para as soluções $t = 2, \dots, v_i - 1$
 - 6 $d_{I_t^m} = d_{I_t^m} + \frac{f_k^{(I_{t+1}^m)} - f_k^{(I_{t-1}^m)}}{f_k^{\max} - f_k^{\min}}$
-

f_k : valor da função objetivo k

I^m : lista ordenada decrescentemente por f_k das soluções em F_i

$d_{I_t^m}$: distância de multidão da t -ésima solução em I^m

$f_k^{(I_{t+1}^m)}$ e $f_k^{(I_{t-1}^m)}$: f_k dos vizinhos da t -ésima solução em I^m

f_k^{\max} e f_k^{\min} : máximo e mínimo global para o f_k

3.5 Sistemas Fuzzy Genéticos

A construção automática de Sistemas Fuzzy (SF) pode ser considerada como um processo de otimização, ou busca no espaço de soluções potenciais. A estrutura do código genético e as características funcionais dos AG os tornam candidatos apropriados para incorporar qual-

quer informação conhecida ou definida previamente, ou seja, é fácil incorporar informações coletadas de especialistas à estrutura dos cromossomos em vários domínios. Recentemente estas vantagens têm sido usadas para estender o uso de AG para o desenvolvimento de uma ampla gama de abordagens para a construção de SF.

Uma dessas vantagens é a possibilidade de incorporar facilmente conhecimento prévio, na forma de variáveis linguísticas, parâmetros de funções de pertinência *fuzzy*, regras *fuzzy*, número de regras, etc., ao processo de construção genética. Assim, durante as últimas duas décadas, aproximadamente, vários artigos e aplicações combinando conceitos *fuzzy* e AG surgiram na literatura e existe uma preocupação crescente sobre a integração entre eles (CORDÓN et al., 2004). Em particular, um grande número de trabalhos explora o uso de AG para o projeto automático de SF. Estas abordagens receberam o nome geral de Sistemas *Fuzzy* Genéticos (SFG).

Este trabalho aborda um tipo particular de SFG, que utiliza AG para gerar os chamados Sistemas *Fuzzy* Baseados em Regras (SFBR), em que um AG é empregado para aprender ou otimizar diferentes componentes de um SFBR (CORDÓN et al., 2004; CASILLAS; CARSE; BULL, 2007; CORDÓN et al., 2007; HERRERA, 2008).

O termo Sistemas *Fuzzy* Genéticos é utilizado para denominar um Sistema *Fuzzy* combinado com um processo de aprendizado baseado em AG. A classe dos SFG inclui as Redes Neurais *Fuzzy* Genéticas, os Algoritmos de Agrupamento *Fuzzy* Genéticos e os SFG Baseados em Regras (SFGBR). As abordagens de SFGBR se baseiam em dois aspectos básicos:

- Qual componente da Base de Conhecimento será otimizada pelos AG - a Base de Regras (BR), a Base de Dados (BD) ou a Base de Conhecimento (BC) completa;
- Qual classe de problemas será abordada: adaptação de elementos da base de conhecimento já definidos, ou a construção de partes de base de conhecimentos sem conhecimento prévio.

Existem dois grupos que combinam as abordagens *fuzzy* e genética para a geração de bases de conhecimento: a Adaptação Genética, que ajusta componentes da base de regras, e a Construção Genética, que constrói os componentes da base de conhecimento. Baseado nestes dois grupos, Herrera (2008) propôs uma taxonomia, apresentando exemplos de cada grupo, conforme mostrado na Figura 3.8 e discutido a seguir.

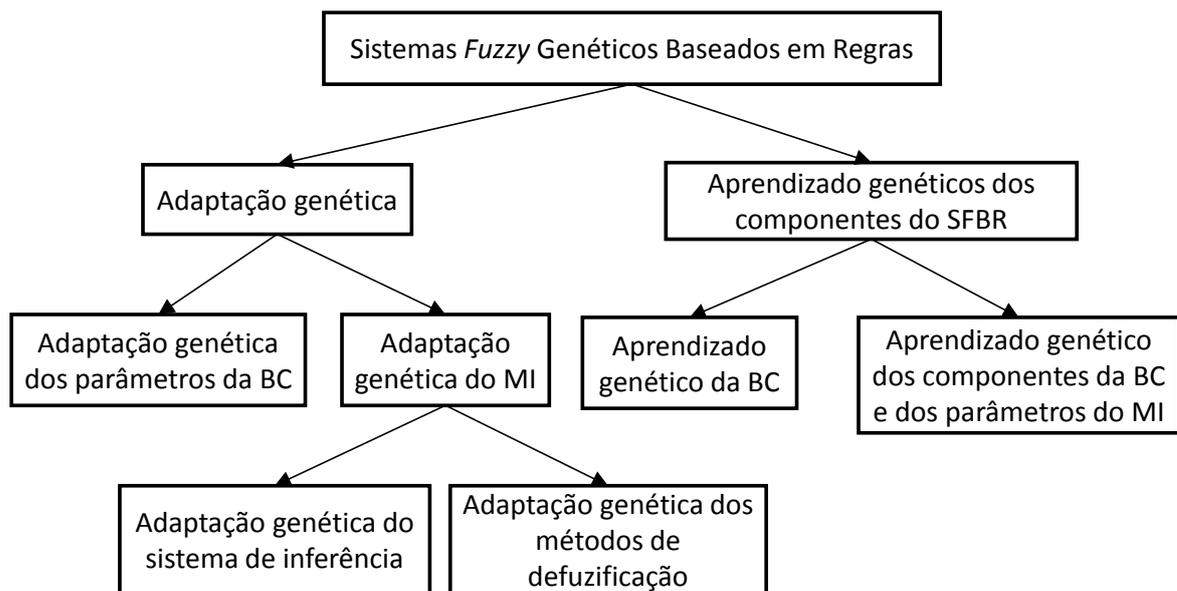


Figura 3.8: Classificação dos Sistemas *Fuzzy Genéticos*. Adaptado de (HERRERA, 2008).

3.5.1 Adaptação Genética

Neste grupo estão incluídos os métodos que utilizam AG para, a partir de uma base de dados ou base de regras já existente, aperfeiçoar o desempenho do sistema fazendo ajuste ou adaptação de uma ou mais partes da BC ou do Mecanismo de Inferência (MI). Esses métodos podem ser subdivididos em grupos de acordo com o enfoque adotado. A seguir são brevemente apresentados esses métodos:

- **Adaptação genética dos parâmetros da BC**

Neste grupo encontram-se os métodos que focalizam a sintonia de conjuntos *fuzzy* (KISSI et al., 2003; HONG et al., 2009; ZHANG; WANG; ZHANG, 2009). Nesses métodos a BR e a BD são previamente definidas, manualmente ou por um processo automático. A BR permanece fixa e a BD é ajustada pelo AG, por meio da alteração dos parâmetros das funções de pertinência de todas as partições *fuzzy* envolvidas no problema. O número de termos linguísticos em cada partição permanece fixo desde o início do processo.

- **Adaptação genética do sistema de inferência**

O principal alvo desta abordagem é obter uma alta cooperação entre as regras *fuzzy*, para aumentar a precisão nos modelos *fuzzy*, sem perder a interpretabilidade linguística, tal como buscar a melhor t-norma utilizada na inferência de um SFBR para aumentar a precisão. Em (ALCALÁ-FDEZ et al., 2007; CROCKETT; BANDAR; MCLEAN, 2007) se pode encontrar propostas nesta área, focadas em problemas de regressão e classificação.

- **Adaptação genética do método de defuzificação**

A técnica de defuzificação mais usada na prática consiste em aplicar a função de defuzificação para cada conjunto *fuzzy* das regras disparadas e, depois, calcular o valor de defuzificação por um operador de média ponderada. Esta forma de cálculo introduz a possibilidade de usar funções de média baseadas em parâmetros, e a utilização de AG pode permitir adaptar o método de defuzificação. Em (KIM; CHOI; LEE, 2002) pode-se encontrar uma proposta nesta área.

3.5.2 Aprendizado Genético dos Componentes dos SFBR

Estão incluídos no grupo do aprendizado genético dos componentes dos SFBR os métodos de aprendizado genético da BC e os métodos baseados em um modelo híbrido entre o aprendizado genético da BC e a adaptação genética do MI. Esse modelo híbrido tenta encontrar uma alta cooperação entre os parâmetros do MI e o aprendizado dos componentes da BC, incluindo-os em um processo de aprendizado simultâneo. Uma proposta desse modelo pode ser encontrada em (MARQUEZ; PEREGRIN; HERRERA, 2007).

Os métodos de aprendizado genético da BC utilizam AG para construir, gerar ou projetar efetivamente um ou mais componentes da BC. Esse grupo de métodos gerou o maior número de pesquisas e pode ainda ser subdividido em quatro subgrupos ou abordagens, brevemente apresentados a seguir.

- **Aprendizado genético da BR**

Essa abordagem estuda a construção das regras *fuzzy* considerando um conjunto de funções de pertinência já definido e fixo. Usualmente a definição da BD é feita escolhendo-se um número de valores linguísticos para as variáveis e distribuindo uniformemente os valores, que são rótulos dos conjuntos *fuzzy* no domínio de cada variável (ABADEH; HABIBI; LUCAS, 2007; EVSUKOFF et al., 2009; SÁNCHEZ; COUSO; CASILLAS, 2009; XIONG, 2009). No aprendizado genético de regras, quatro abordagens são comumente utilizadas (CORDÓN et al., 2004): (1) abordagem de Pittsburgh, em que cada cromossomo representa uma BR completa, na qual a BR final é o melhor cromossomo obtido (SMITH, 1980); (2) abordagem de Michigan, em que cada regra é codificada como um cromossomo e a BR final é a união de todos os cromossomos da melhor população (HOLLAND; REITMAN, 1978); (3) abordagem iterativa, em que cada regra é codificada como um cromossomo adicionada à BR final de forma iterativa a cada execução do AG (GONZÁLEZ; PÉREZ, 1999); (4) e a abordagem Cooperativa-Competitiva, em que a população inteira, ou um subconjunto

desta população, codifica a BR e os cromossomos competem e cooperam simultaneamente (GREENE; SMITH, 1993).

- **Seleção genética de regras**

Ocasionalmente, pode-se ter um grande número de regras extraídas por meio de um método de mineração de dados. Uma BR com um número excessivo de regras dificulta o entendimento do comportamento do SFBR. Diferentes tipos de regras podem ser encontrados em um conjunto de regras *fuzzy*: regras irrelevantes, redundantes, erradas e conflitantes, que comprometem o desempenho do SFBR. Para tratar este problema, pode-se utilizar um processo de seleção de regras genético para obter um subconjunto de regras a partir de um conjunto prévio de regras *fuzzy* mediante a seleção de algumas dessas regras (KUWAJIMA; ISHIBUCHI; NOJIMA, 2008).

- **Aprendizado genético da BD**

Existem outras formas para gerar a BC que consideram dois diferentes processos que derivam ambos os componentes, a BD e a BR. Na primeira possibilidade, definida como aprendizado genético a priori da BD, um processo de geração da BD permite aprender a forma da função de pertinência dos conjuntos *fuzzy*. Este processo de geração da BD pode usar uma medida para avaliar a sua qualidade; em seguida, um método de geração de regras é utilizado. A segunda possibilidade é considerar um processo de aprendizado genético embutido, onde a geração da BD envolve um aprendizado da BR, ou seja, cada vez que uma BD é obtida pelo processo de geração, o método de geração da BR é usado para derivar as regras; um tipo de medida de erro é então utilizado para validar a BC. Algumas propostas sobre o aprendizado genético da BD podem ser encontradas em (PIRES; CARMARGO, 2004; HONG et al., 2009). A Figura 3.9 ilustra as subdivisões do aprendizado genético da BD.

- **Aprendizado genético simultâneo dos componentes da BC**

Esta abordagem tem como objetivo aprender os dois componentes da BC simultaneamente (B. Filipic, 1996). Seguindo esta abordagem é possível gerar uma melhor definição do SFBR, mas a necessidade de lidar com um grande espaço de busca torna o processo difícil e lento.

3.6 Sistemas *Fuzzy* Genéticos Multiobjetivo

Os AGMO, apresentados na Seção 3.4, constituem uma das áreas mais ativas de pesquisa no campo da Computação Evolutiva (HERRERA, 2008). Estes algoritmos são capazes de encontrar

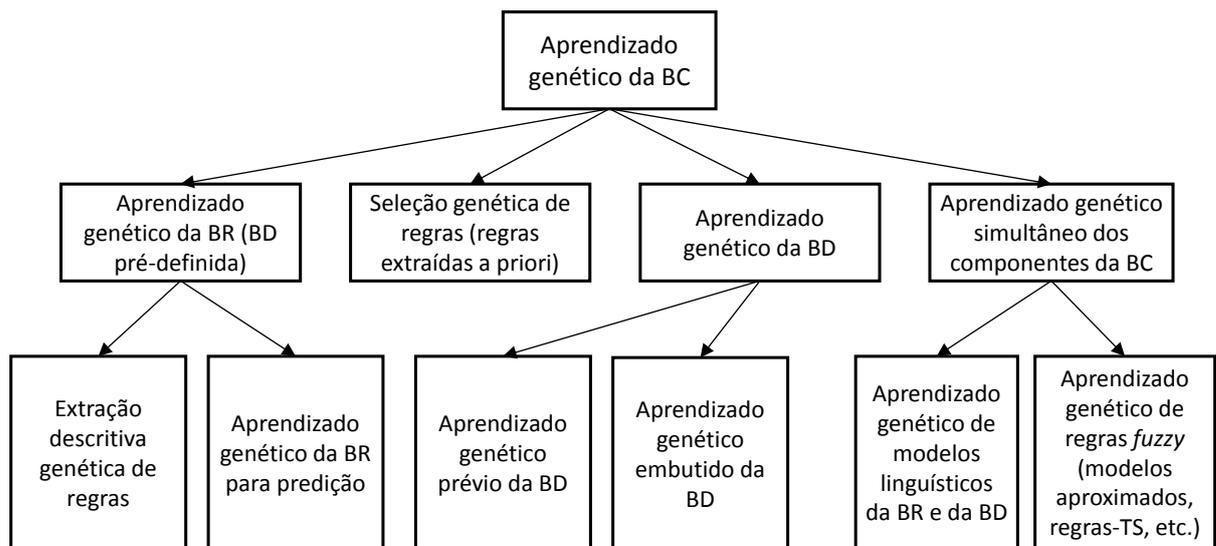


Figura 3.9: Aprendizagem genética simultânea dos componentes da BC. Adaptado de (HERRERA, 2008).

um conjunto de soluções não dominadas em uma única execução.

O projeto de sistemas *fuzzy* estabelece dois requisitos para o modelo a ser definido:

- **Acuidade** (ou precisão), que é a capacidade de representar o sistema real com fidelidade;
- **Interpretabilidade** (ou clareza), que é a capacidade de expressar o comportamento do sistema real de uma maneira compreensível ao ser humano.

Entretanto, obter altos graus de interpretabilidade e acuidade são objetivos contraditórios e, na prática, um dos dois objetivos frequentemente prevalece sobre o outro. Esta é uma dificuldade frequentemente encontrada em SFBR e, no decorrer dos últimos anos, reconheceu-se não ser apropriado negligenciar qualquer um desses objetivos, fato este que originou a tendência de procurar um balanceamento entre interpretabilidade e acuidade no projeto de SF.

Por serem objetivos contraditórios, a busca por alta interpretabilidade e alta acurácia na geração de SFBR tem sido grandemente explorada por meio do uso de AGMO, originando os Sistemas *Fuzzy* Genéticos Multiobjetivo (SFGMO).

Fazzolari et al. (2013) apresentam uma taxonomia para os trabalhos que utilizam os SFGMO. Essa taxonomia é dividida em dois grupos principais. O primeiro grupo aborda os trabalhos de acordo com os objetivos a serem otimizados. O segundo grupo aborda os trabalhos de acordo com o componente do SFBR a ser otimizado. A Figura 3.10 mostra esta taxonomia. A linha tracejada divide os dois principais grupos.



Figura 3.10: Classificação dos SFGMO. Adaptado de (FAZZOLARI et al., 2013).

Nesta taxonomia, a área classificada como *Geração de SFBR com balanceamento entre acurácia e interpretabilidade* é a área de interesse deste trabalho.

Nos últimos anos, diversos trabalhos têm dado atenção ao problema de interpretabilidade em SFBR (CORDÓN, 2011). Alguns destes trabalhos têm buscado encontrar formas de medir a interpretabilidade de SF de forma a encontrar um índice de interpretabilidade amplamente aceitável pela comunidade (ALONSO; MAGDALENA; GONZÁLEZ-RODRÍGUEZ, 2009).

Encontrar o balanceamento entre a acurácia e a interpretabilidade de um SF é uma tarefa difícil já que estes objetivos são conflitantes. Desta maneira, os AGMO são utilizados para encontrar um conjunto de soluções factíveis com diferentes níveis de balanceamento entre a acurácia e a interpretabilidade (CANNONE; ALONSO; MAGDALENA, 2011).

A seguir são descritos alguns trabalhos relacionados à tarefa de balancear a acurácia e a interpretabilidade do SFBR.

3.6.1 Adaptação Genética Multiobjetivo

Neste grupo estão incluídos os trabalhos que utilizam AGMO para, a partir de uma base de dados ou base de regras já existente, ajustar ou adaptar uma ou mais partes da BC. Esses métodos podem ser subdivididos em grupos de acordo com o enfoque adotado.

- **Ajuste da BD:**

Os trabalhos com foco no ajuste da BD fazem uso dos AGMO para ajustar os parâmetros das funções de pertinência da BD. A BR utilizada permanece fixa até o término do AGMO. Em Gacto, Alcalá e Herrera (2009), os autores propuseram um AGMO para encontrar a granularidade de cada variável e ajustar os parâmetros das funções de pertinência com o intuito de tratar problemas de alta dimensionalidade. Di Nuovo e Catania (2009) e Pulkkinen e Koivisto (2008) utilizam um AGMO para otimizar a BD buscando o equilíbrio entre a acurácia e a interpretabilidade do SFBR.

- **Ajuste dos parâmetros do mecanismo de inferência:**

Uma outra abordagem utilizada para aumentar a acurácia dos SFBR é o ajuste dos parâmetros dos mecanismos de inferência (ALCALÁ-FDEZ et al., 2007). Em (MÁRQUEZ; MÁRQUEZ; PEREGRÍN, 2008) o NSGA-II e o SPEA2 foram alterados a fim de ajustar os operadores do mecanismo de inferência juntamente com as regras, obtendo BRs mais compactas e precisas. Em (MARQUEZ; MARQUEZ; PEREGRIN, 2012) um AGMO é proposto para ajustar os operadores de conjunção do mecanismo de inferência.

3.6.2 Aprendizado Genético Multiobjetivo dos Componentes dos SFBR

Pertencem ao grupo de aprendizagem genética dos componentes dos SFBR os métodos que utilizam AGMO para selecionar regras a partir de um conjunto de regras previamente extraídas dos conjuntos de dados; os métodos que utilizam AGMO para gerar BRs a partir de um conjunto de dados; e os métodos que utilizam AGMO para gerar a BC completa. Estes métodos são brevemente apresentados a seguir.

- **Aprendizado por seleção de regras**

Para tratar o problema de dimensionalidade, Ishibuchi e Yamamoto (2004b) utilizam um AGMO para extrair regras *fuzzy* a partir de um conjunto de dados levando em consideração três objetivos: aumentar a acurácia, minimizar o número de regras e minimizar o número de antecedentes das regras. Em (GACTO; ALCALA; HERRERA, 2011) os autores usaram AGMO para seleção de regras *fuzzy* para a geração de BR compactas e com boa acurácia de controladores.

- **Aprendizado da BR**

Em (ANTONELLI et al., 2008) os autores apresentam um AGMO para aprender a BR e a granularidade da BD simultaneamente. Em (REY et al., 2013), os autores usam o conceito de *transformação ortogonal* que fornece ao AGMO informações a respeito de regras

candidatas, auxiliando-o a selecionar as melhores regras e gerar BRs mais compactas. Recentemente, (JEE et al., 2014) apresentaram uma proposta para geração de SFBR a partir de seleção de regras.

- **Aprendizado dos componentes da BC**

Em (CORDON et al., 2001), os autores propuseram um método para a seleção de atributos e aprendizagem da BD para obterem um SFBR com BR compactas, compreensivas e precisas. O AGMO tinha dois objetivos: minimizar o erro de classificação e obter uma BD interpretável, penalizando os classificadores *fuzzy* com grande quantidade de atributos selecionados e alta granularidade. O segundo objetivo é expresso como o produto de atributos selecionados e suas granularidades médias.

Em (ALCALÁ et al., 2009a), foi proposta uma técnica que aprende simultaneamente a BR e a BD de modelos de regressão *fuzzy*. Dois AGMO são utilizados para gerarem um conjunto de SFBR com diferentes taxas de balanceamento entre a acurácia e a interpretabilidade. O AGMO aprende a BR e os parâmetros da função de pertinência simultaneamente.

3.7 Considerações finais

Neste capítulo foi apresentada uma breve descrição sobre algoritmos genéticos, assim como uma breve introdução sobre as outras técnicas abrangidas pela computação evolutiva. Foi apresentado também como os AG podem ser utilizados para resolver problemas de otimização multiobjetivo, definidos como Algoritmos Genéticos Multiobjetivo (AGMO), mudando-se principalmente a forma de cálculo da aptidão de cada solução, baseada na dominância das soluções. Neste contexto, foi apresentado o funcionamento do NSGA-II, um dos AGMO mais conhecidos.

Também foram discutidos os SFG. Os SFG têm sido amplamente investigados nas últimas duas décadas, período em que surgiram as mais variadas abordagens para tratamento da questão de aprendizado e otimização de sistemas *fuzzy* com o uso de AGs. Mais recentemente, o uso de AGMO tem sido muito explorado na geração de SFBR com foco na questão do balanceamento entre a acurácia e a interpretabilidade. O sucesso obtido em muitos casos indica que esse é um ramo promissor na pesquisa da geração automática de bases de conhecimento.

No próximo capítulo será apresentada a proposta para geração genética multiobjetivo de SFBR.

Capítulo 4

GERAÇÃO GENÉTICA MULTIOBJETIVO DE SFBR COM FOCO NA DISTRIBUIÇÃO DAS SOLUÇÕES

4.1 Considerações Iniciais

Como exposto no Capítulo 3, a preocupação com a distribuição das soluções na fronteira de Pareto é um tópico de interesse para a comunidade científica que trabalha com os AGMOs. Este tópico torna-se importante quando se observa que as soluções obtidas por meio dos AGMOs podem fornecer poucas opções ao usuário quanto ao equilíbrio entre os objetivos a serem otimizados, ou seja, um conjunto de soluções pode convergir para uma área restrita do espaço de busca, onde se pode encontrar soluções pouco diversificadas.

Para abordar este problema, propõe-se neste trabalho um método para a geração genética de SFBR que possibilita uma boa distribuição das soluções ao longo da fronteira de Pareto, tendo como classe de problemas abordada a de classificação de padrões multidimensionais.

4.2 Metodologia para Geração de Base de Conhecimento *Fuzzy*

A metodologia para a geração genética multiobjetivo de SFBR com foco na distribuição das soluções, proposta neste trabalho, é composta por duas etapas:

- **Seleção de regras:** nesta etapa um AGMO é utilizado com o intuito de gerar regras *fuzzy* capazes de representar o conhecimento existente no conjunto de dados. Cada variável é granularizada em 3 conjuntos *fuzzy* igualmente espaçados que permanecem fixos durante o processo genético desta etapa;
- **Otimização da BD:** nesta etapa um AGMO é utilizado para otimizar os conjuntos *fuzzy*

utilizados na etapa anterior.

A Figura 4.1 ilustra de maneira ampla os processos envolvidos na etapa 1 da metodologia.

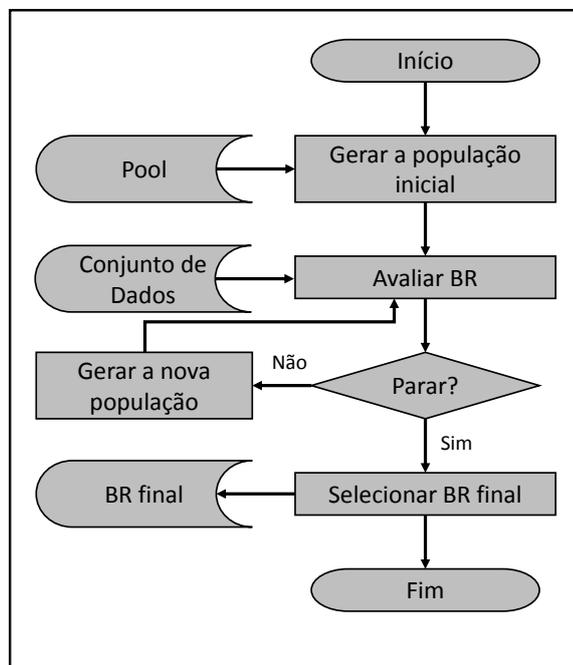


Figura 4.1: Etapa1 - Seleção de regras

Inicialmente, é gerada uma população de indivíduos, em que cada indivíduo armazena índices de regras selecionadas de um *pool* de regras e representa uma BR completa. Estes índices são escolhidos aleatoriamente, não sendo permitida a escolha de índices repetidos para o mesmo indivíduo a fim de evitar redundâncias.

Enquanto o critério de parada não é satisfeito, novas populações serão geradas por meio dos operadores genéticos do AGMO.

Por fim, quando se atinge o critério de parada (número máximo de gerações), a solução central da fronteira de Pareto é escolhida para a etapa 2.

Escolhe-se a solução central da fronteira por ser esta a solução que mantém o maior equilíbrio entre a Taxa de Classificação Correta (TCC) e a complexidade da BR.

A etapa 2 da metodologia é ilustrada na Figura 4.2. Nesta etapa, a BR encontrada na etapa anterior permanecerá fixa, enquanto que a BD será otimizada por um AGMO.

O primeiro indivíduo da população inicial é a BD utilizada na etapa anterior; os demais indivíduos são gerados de forma aleatória e, após a avaliação da população atual, novas populações serão geradas por meio dos operadores genéticos do AGMO até que a condição de parada seja satisfeita.

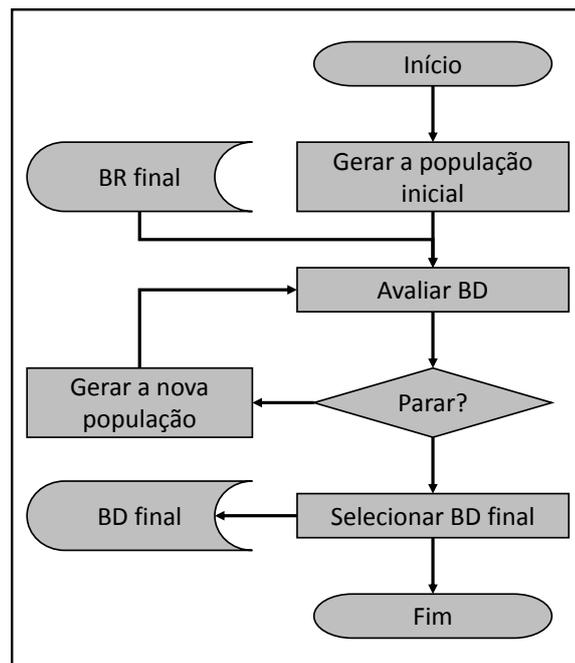


Figura 4.2: Etapa 2 - Otimização da BD

Após atingir a condição de parada do AGMO desta etapa, a solução central da fronteira da última geração do AGMO será a solução escolhida.

Com o intuito de atacar o problema da dispersão das soluções não dominadas, neste trabalho é proposta uma abordagem de seleção de indivíduos para comporem a geração seguinte a cada passo do AGMO que visa a manter uma boa distribuição das soluções na fronteira de Pareto. Para isso, a cada geração do AGMO é avaliada a dispersão das soluções ao longo da fronteira de Pareto. A seleção dos indivíduos é, então, feita com base nessa avaliação, com o objetivo de gerar soluções equidistantes.

A abordagem proposta parte do pressuposto de que as soluções não dominadas precisam estar equidistantes para que a dispersão destas soluções seja equilibrada.

Uma vez encontradas soluções não dominadas equidistantes, pode-se dizer que a distância entre estas soluções é a distância ideal para as soluções da fronteira de Pareto da geração atual do AGMO.

A partir do momento em que se conhece a distância ideal entre as soluções para uma determinada fronteira, pode-se medir quão bem distribuídas estão estas soluções e, conseqüentemente, avaliar a sua dispersão.

O cálculo do espaçamento ideal, o índice de distribuição das soluções e o método de escolha da solução central serão discutidos nas próximas seções.

4.3 Cálculo do Espaçamento Ideal entre as Soluções

A cada geração de um AGMO, encontra-se um conjunto de soluções não dominadas S_t que definem a fronteira de Pareto na geração t do AGMO. A distância entre as soluções de uma mesma fronteira, bem como o formato e o comprimento de cada fronteira, variam de geração para geração.

Para encontrar o espaçamento ideal entre as soluções não dominadas de uma determinada fronteira, é necessário primeiramente conhecer o comprimento desta fronteira. Para tanto, deve-se inicialmente encontrar a função linear f_{ij}^k que intercepta uma solução não dominada s_i e a solução não dominada vizinha s_j . Esta função f_{ij}^k que intercepta as soluções não dominadas s_i no ponto (x_i, y_i) e s_j no ponto (x_j, y_j) é definida pela equação 4.1:

$$f_{ij}^k(x) = a_k x + b_k \quad (4.1)$$

a qual:

- a_k é o coeficiente angular da reta definida pela função k calculada de acordo com a equação 4.2, sendo $k = 1, \dots, nS - 1$ e nS é o número de soluções não dominadas da geração atual;
- b_k é o intercepto do eixo y no plano cartesiano;
- x_i e x_j representam a taxa de interpretabilidade das soluções i e j , respectivamente;
- y_i e y_j representam a taxa de classificação correta das soluções i e j , respectivamente.

$$a_k = \frac{y_j - y_i}{x_j - x_i} \quad (4.2)$$

Após encontrar todas as $k = \{1, 2, \dots, nF\}$ funções, com $nF = nS - 1$, calcula-se o comprimento l_k da função f_{ij}^k entre os pontos (x_i, y_i) e (x_j, y_j) pela equação 4.3.

$$l_k = g_k(x_j) - g_k(x_i), \quad j > i \quad (4.3)$$

sendo g_k definida por:

$$g_k = x \sqrt{1 + a_k^2} \quad (4.4)$$

Desta forma, o comprimento L da fronteira é calculado conforme a equação 4.5.

$$L = \sum_{k=1}^{nF} l_k \quad (4.5)$$

A partir do momento em que o comprimento L da fronteira é conhecido, o espaçamento ideal E entre as soluções não dominadas desta fronteira pode ser calculado utilizando-se a equação 4.6.

$$E = \frac{L}{nF} \quad (4.6)$$

O espaçamento E é a medida que será utilizada como referência para calcular a coordenada dos pontos ideais da fronteira de Pareto atual onde deveria existir uma solução para que o espaçamento entre as soluções seja igual, encontrando desta forma uma distribuição equilibrada das soluções ao longo da fronteira de Pareto. Para calcular a coordenada dos pontos ideais é proposto o Algoritmo 4.1.

4.4 Índice de Distribuição

Em alguns trabalhos (HÁJEK; SZÖLLÖS; VSÍSTEK, 2010; Da Ronco; BENINI, 2013; LAUMANN et al., 2002; TOFFOLO; BENINI, 2003), os autores fazem a análise da distribuição das soluções ao longo da fronteira de Pareto de forma visual observando os pontos plotados em gráficos. Muitos dos trabalhos em que a análise da distribuição das soluções é feita de forma visual, aplicam o AGMO para otimizar funções cujo formato simples da fronteira de Pareto não dificulta a observação visual. Como exemplos de funções que apresentam esta característica tem-se os problemas de otimização matemática introduzidos por Deb et al. (2002b).

A Figura 4.3 mostra um exemplo em que avaliar a distribuição das soluções de forma visual não é difícil. Na figura, pode-se observar que a distribuição das soluções não dominadas (círculos pretos) ao longo da fronteira aparentemente é uniforme.

Entretanto, no caso da Figura 4.4, ter-se-á mais dificuldade de avaliar se as soluções não dominadas estão uniformemente distribuídas. Isto porque as curvas nos pontos de inflexão da função que define a fronteira de Pareto podem confundir o observador, levando-o à uma análise imprecisa e incorreta.

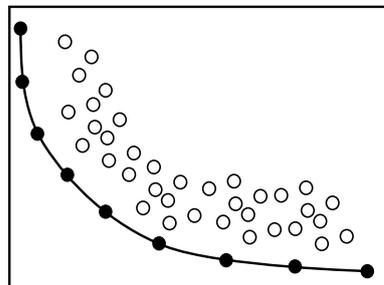
A fim de obter uma análise mais precisa, algumas medidas têm sido empregadas para verificar a distribuição das soluções na fronteira de Pareto. Como exemplo, tem-se a diversidade

Algoritmo 4.1: EncontraPontosIdeais**Entrada:** E : espaçamento ideal. S_t : matriz de dimensão $nS \times 2$ que representa as coordenadas das soluções da fronteira de Pareto da geração t .**Saída:** SI_t : matriz de dimensão $nS \times 2$ que representa as coordenadas dos pontos ideais na fronteira de Pareto da geração t .**1 Início**

```

2    $cP := 0$ ; // Comprimento percorrido
3    $cFAtual := 0$ ; // Comprimento da função atual
4    $trocaAtual := 2$ ; // Comprimento atual
5    $pAtual := 2$ ; // Ponto real atual
6   Para  $k = 1$  até  $nF$  Faça
7        $cFAtual := g_k(S_t(pAtual, 1)) - g_k(S_t(pAtual - 1, 1))$ ;
8       Enquanto  $cP + cFAtual > E$  &&  $trocaAtual == nS$  Faça
9           Se  $cP <> 0$  Então
10               $xIdeal := |g_k(xIdeal) - g_k(S_t(pAtual - 1, 1)) - |(E - cP)||$ ;
11              Senão
12                   $xIdeal := |g_k(xIdeal) - g_k(SI_t(trocaAtual - 1, 1)) - E|$ ;
13              Fim-Se
14               $yIdeal := a_k * xIdeal + b_k$ ;
15               $SI_t(trocaAtual, 1) := xIdeal$ ;
16               $SI_t(trocaAtual, 2) := yIdeal$ ;
17               $cFAtual := g_k(S_t(pAtual, 1)) - g_k(SI_t(trocaAtual, 1))$ ;
18               $trocaAtual := trocaAtual + 1$ ;
19               $cP := 0$ ;
20          Fim-Enquanto
21           $cP := cP + cFAtual$ ;
22           $pAtual := 0$ ;
23      Fim-Para
24 Fim

```

**Figura 4.3:** Fronteira com uma única inflexão

genética, que mede a variedade e a distribuição de indivíduos, definida por Solow e Polasky (1994) como:

$$D(P) = \mathbf{eM}^{-1}\mathbf{e}^T \quad (4.7)$$

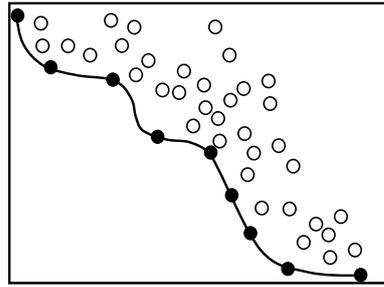


Figura 4.4: Fronteira com muitas inflexões

Nesta equação, \mathbf{e} é um vetor de n posições contendo o valor 1 e \mathbf{M} é uma matriz $n \times n$ definida pela equação 4.8, em que d é a distância euclidiana entre os pontos \mathbf{x}_i e \mathbf{x}_j , e o parâmetro θ é definido arbitrariamente.

$$m_{ij} = \exp(-\theta \cdot d(\mathbf{x}_i, \mathbf{x}_j)), \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, n; \quad \theta > 0 \quad (4.8)$$

A ideia geral desta medida é avaliar a diversidade das soluções no espaço de decisão e foi usada por Ishibuchi et al. (2012) para medir a diversidade das soluções no espaço de busca e na fronteira de Pareto obtida. Esta é uma medida que reflete principalmente a densidade de grupos de soluções presentes no espaço de busca e não avalia a uniformidade da distribuição das soluções na fronteira de Pareto.

Ao definir o parâmetro θ , o espaço de interesse na fronteira de Pareto diminui, e, conseqüentemente a diversidade diminui. Mesmo que as soluções estejam igualmente espaçadas naquela região, a diversidade é menor do que o mesmo número de soluções igualmente espaçadas ao longo de toda a fronteira. Isto se deve ao fato de esta medida utilizar todo o espaço de busca como referência, e não apenas a função que define a fronteira de Pareto.

Dois outras medidas amplamente utilizadas na literatura para medir a diversidade das soluções na fronteira de Pareto são: hipervolume (AZEVEDO; ARAUJO, 2011) e hipercubo (ADRA; FLEMING, 2011). A desvantagem destas medidas é a utilização da distância euclidiana como base. Considerando o fato de que, dependendo do problema, a fronteira de Pareto pode ser definida por uma função com muitos pontos de inflexão, usar um segmento de reta entre dois pontos para medir a distância entre eles faz com que a informação contida no ponto de inflexão seja perdida.

Considerando que a melhor distribuição das soluções na fronteira é aquela cujas soluções

estão sobre os pontos ideais, define-se o seguinte índice de distribuição ID :

$$ID = \frac{1}{L} \sum_{\substack{i=1 \\ j=i+1}}^{nS-1} |d_{ij} - E| \quad (4.9)$$

Uma vez encontrados os pontos ideais na fronteira de Pareto na geração t , pode-se medir quão distantes as soluções S_t estão destes pontos. O espaçamento E contém parte das informações contidas nos pontos de inflexão da função que define a fronteira de Pareto, fornecendo então uma avaliação mais precisa da uniformidade da distribuição das soluções ao longo da fronteira de Pareto.

Como exemplo do uso deste índice, considere-se a Figura 4.5.

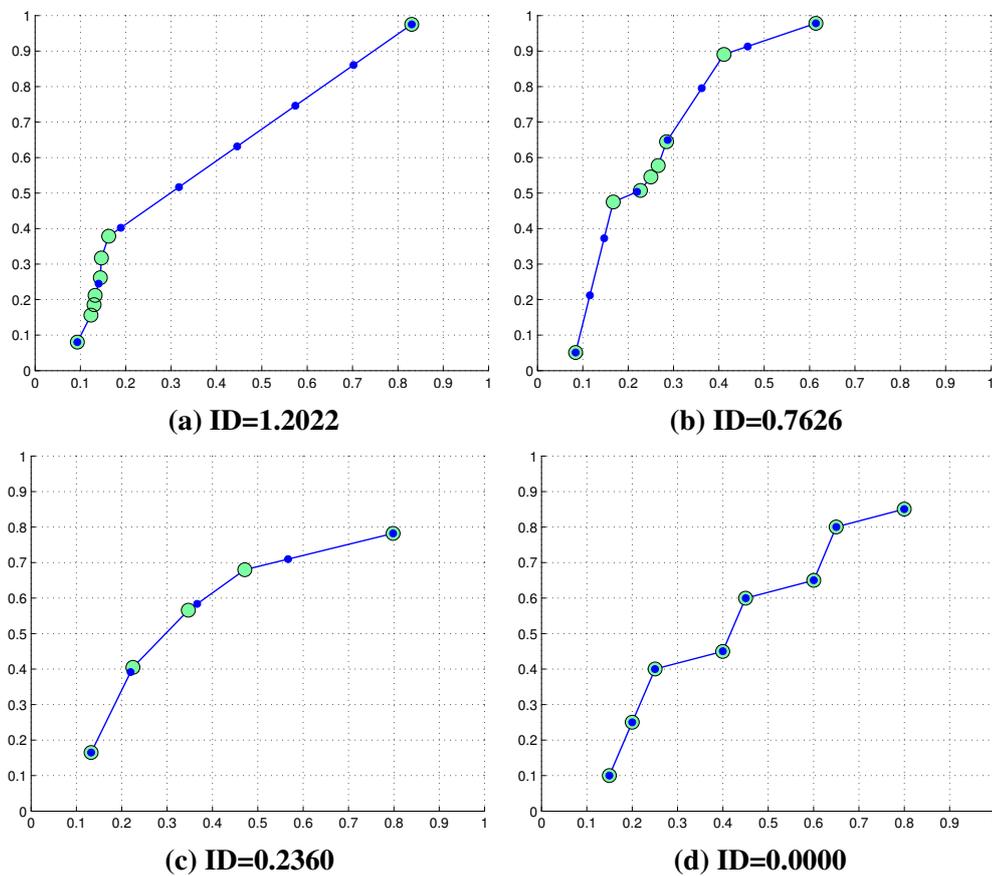


Figura 4.5: Exemplos do uso do Índice de Distribuição

Nesta figura, os círculos verdes são as soluções não dominadas encontradas, e os pontos azuis são os pontos ideais. Pode-se ver que quanto mais próximas as soluções encontradas estiverem dos pontos ideais, menor será o ID e mais equilibrada estas soluções estarão.

4.5 Escolha da Solução Central

Após atingir a condição de parada, o AGMO fornece um conjunto de soluções não dominadas. A partir deste ponto, é necessário decidir qual das soluções não dominadas será retornada ao usuário. Esta é uma tarefa importante, pois entra como fator de decisão o equilíbrio entre os objetivos.

Uma das formas de fazer esta escolha, é por meio de um especialista (DELGADO; Von Zuben; GOMIDE, 2002; CORDON et al., 2001). Dentre as opções fornecidas pelo AGMO, o especialista irá optar pela solução que melhor se adeque ao problema. A desvantagem desta abordagem é a eventual inexistência de um especialista que possa ser consultado. Além disso, em metodologias em que há mais de um estágio, o sistema fica ocioso até que o especialista faça a escolha. Só então o sistema pode prosseguir para a próxima etapa. Por conta disso, métodos automáticos de escolha são desejados.

Criar métodos automáticos envolve a tarefa de encontrar a solução que mais bem equilibra os objetivos, o que nem sempre é possível dependendo da diversidade das soluções encontradas (ISHIBUCHI; NOJIMA, 2013).

Um exemplo de método automático de escolha é o método baseado na inflexão da função que define a fronteira. O problema deste método de escolha é a necessidade de dar um peso maior para um dos objetivos quando há mais de um ponto de inflexão. Se existir um ponto de inflexão e este ponto de inflexão não estiver no meio da fronteira de Pareto, um dos objetivos será beneficiado.

O método automático mais utilizado consiste em medir a distância euclidiana entre as duas soluções mais extremas e encontrar o ponto médio (PM) desta distância. A solução que mais se aproximar deste ponto será a solução escolhida (CÁRDENAS; CAMARGO, 2012). A principal desvantagem deste método, semelhantemente à questão discutida na seção 4.4 sobre a dispersão das soluções na fronteira, é o fato de este método não considerar as informações contidas nas inflexões da função que define a fronteira.

Para escolher a solução central, é proposto um método que calcula a coordenada do *Ponto de Equilíbrio da Fronteira (PEF)*. O PEF representa o ponto máximo de equilíbrio entre os objetivos da fronteira da geração atual do AGMO.

Para encontrar o PEF, basta fazer uma adaptação do Algoritmo 4.1, como pode-se observar no algoritmo 4.2.

Pode-se observar na Figura 4.6 uma comparação entre método automático baseado na pro-

Algoritmo 4.2: EncontraPontoDeEquilibrio**Entrada:**

S_t : matriz de dimensão $nS \times 2$ que representa as coordenadas das soluções da fronteira de Pareto da geração t .

Saída:

PEF : coordenada do Ponto de Equilíbrio da Fronteira de Pareto da geração t .

1 Início

```

2    $cP := 0$ ; // Comprimento percorrido
3    $cFAtual := 0$ ; // Comprimento da função atual
4    $pAtual := 2$ ; // Ponto real atual
5    $mL := L/2$ ; //  $L$  representa o tamanho da fronteira
6    $k := 1$ ; // Função matemática atual
7   Enquanto  $k \leq nF$  Faça
8        $cFAtual := g_k(S_t(pAtual, 1)) - g_k(S_t(pAtual - 1, 1))$ ;
9       Se  $cP + cFAtual > mL$  Então
10           $x := 0$ ;
11           $y := 0$ ;
12           $xPEF := |g_k(xIdeal) - g_k(S_t(pAtual - 1, 1)) - |(mL - cP)||$ ;
13           $yPEF := a_k * xPEF + b_k$ ;
14           $k := nF + 1$ ; // Força sair do laço
15      Fim-Se
16       $cP := cP + cFAtual$ ;
17       $pAtual := 0$ ;
18       $k := k + 1$ ;
19  Fim-Enquanto
20 Fim

```

ximidade com o PM entre as soluções extremas da fronteira e o método automático baseado na proximidade com o PEF.

Os pontos em vermelho representam as soluções reais encontradas pelo AGMO. Os pontos azuis representam a posição em que a distribuição é uniforme. O ponto verde indica o PEF e o ponto magenta indica o PM, que está localizado no meio da reta magenta que liga os dois pontos extremos da fronteira. A linha azul é a fronteira de Pareto. Por fim, os círculos azul e magenta representam a solução mais próxima do PEF e do PM, respectivamente.

Considerando que o eixo x é a interpretabilidade e o eixo y a acurácia do sistema, as Figuras 4.6a e 4.6b representam um AGMO em que se deseja maximizar os dois objetivos. Pode-se observar na Figura 4.6a que a solução mais próxima do PEF tem interpretabilidade um pouco inferior à solução mais próxima do PM. Porém, a acurácia é aproximadamente 20% maior na primeira solução. Já na Figura 4.6b a solução mais próxima do PEF possui a acurácia um pouco mais baixa com relação à solução mais próxima do PM. Entretanto, a interpretabilidade é quase 40% maior.

Nas Figuras 4.6c e 4.6d, que ilustram o caso em que o AGMO está buscando minimizar o índice da interpretabilidade e maximizar o índice da acurácia, pode-se observar que algo similar acontece. A solução mais próxima do PEF possui aproximadamente 35% a mais de interpretabilidade do que a solução mais próxima do PM na Figura 4.6c e aproximadamente 65% a mais de acurácia com relação à solução mais próxima do PM. No objetivo em que a solução mais próxima do PEF perde para a solução mais próxima do PM, a diferença é muito pequena quando comparada ao ganho no outro objetivo.

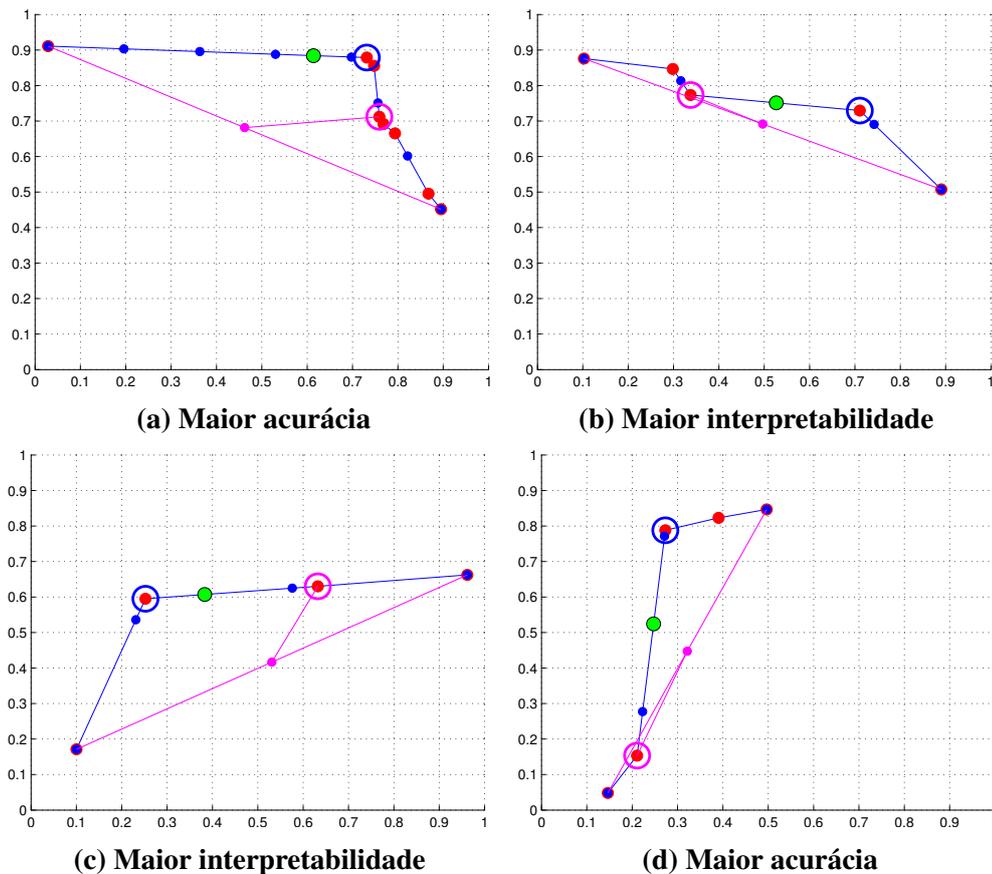


Figura 4.6: Diferença entre a escolha por PM e por PEF

Com isso pode-se ver que o PEF é uma boa medida de referência para a escolha da solução central da fronteira de Pareto, fornecendo desta maneira a solução que apresenta o maior equilíbrio nos objetivos do problema.

4.6 Non-Dominated Sorting Genetic Algorithm Distance Oriented (NSGA-DO)

Uma das principais características do NSGA-II é o conceito de distância de multidão introduzido para manter a diversidade das soluções. Todavia, se o conjunto de soluções não

dominadas F_1 tem um tamanho maior que N (tamanho da população P), o processo de escolher apenas N soluções usando a distância de multidão faz com que algumas soluções sejam perdidas. Considere-se F_1 contendo várias soluções Pareto-ótimas muito próximas e alguma solução distante não Pareto-ótima, mas não dominada no momento. Dado que o cuboide da solução não dominada é maior, esta solução será copiada em $|Pn + 1|$ (sendo n a geração atual) enquanto uma solução Pareto-ótima é eliminada. A Figura 4.7 ilustra esta situação.

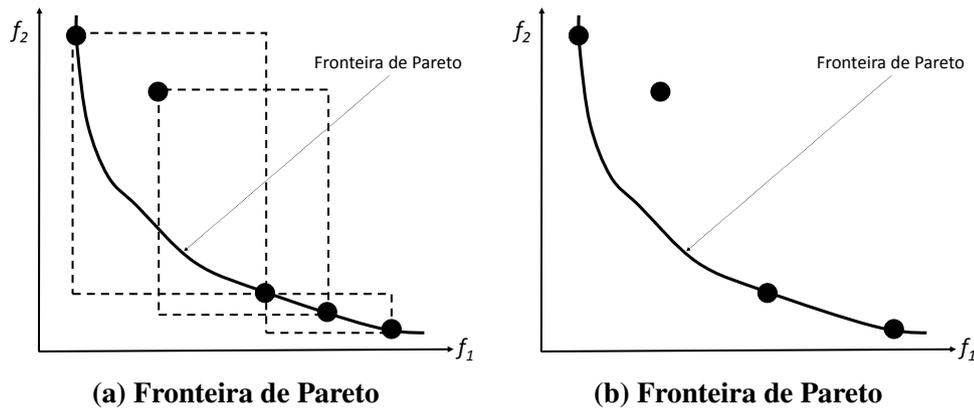


Figura 4.7: Situação em que o algoritmo de corte do NSGA-II erra

Esta situação faz que o NSGA-II possa cair em um ciclo de gerar soluções Pareto-ótimas e não Pareto-ótimas até convergir finalmente a um conjunto de soluções Pareto-ótimas.

Nesta tese proposta, é apresentada a versão do NSGA-II orientada à distância, denominada aqui *Non-Dominated Sorting Genetic Algorithm Distance Oriented (NSGA-DO)*.

O NSGA-DO tem como principal característica orientar as soluções de forma a convergirem para os pontos ideais da fronteira de Pareto, privilegiando as soluções mais próximas a algum ponto ideal da fronteira ao trocar a distância de multidão do NSGA-II pela distância entre uma solução real encontrada e o ponto ideal mais próximo desta solução. Isso faz com que soluções contidas na fronteira de Pareto de uma geração t não sejam trocadas por uma solução não dominada que não pertence à fronteira atual.

A Figura 4.8 ilustra um exemplo da diferença de usar como critério de seleção a solução mais próxima a um ponto ideal em vez da distância de multidão.

Nesta figura, F_2 representa uma fronteira de Pareto e F_1 representa a fronteira de Pareto ótima. Os círculos cinzas representam as soluções encontradas por um AGMO e que formam F_1 e F_2 . Os pontos pretos na fronteira F_1 representam os pontos ideais.

Supondo que tivéssemos que selecionar oito soluções para a próxima geração, o NSGA-II escolheria as soluções das primeiras fronteiras. Assim, todas as soluções na fronteira F_1 seriam

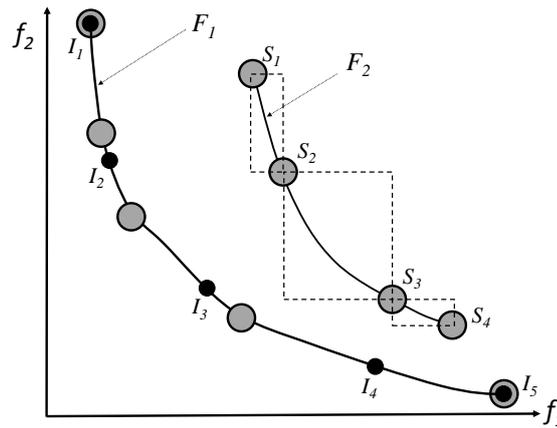


Figura 4.8: Escolha de solução por proximidade ao ponto ideal

selecionadas.

Para completar o total de oito soluções, o NSGA-II usará a distância de multidão para escolher três soluções na fronteira F_2 . As soluções S_1 e S_4 , por estarem nas extremidades, possuem distância de multidão igual a infinito e seriam as primeiras escolhidas. Entre as soluções S_2 e S_3 , a solução S_2 seria escolhida, pois possui distância de multidão maior que S_3 .

Pode-se ver que a distância de multidão fez com que a solução S_3 , a mais próxima de um ponto ideal (ponto I_4), fosse descartada.

As soluções escolhidas pelo NSGA-DO são aquelas mais próximas dos pontos ideais. Para este exemplo, todas as soluções da fronteira F_1 seriam escolhidas. Dentre as soluções da fronteira F_2 , a ordem de escolha seria S_3 , S_4 e S_2 . Com S_3 e S_4 pertencendo a população próxima geração aumenta a quantidade de soluções próximas aos pontos ideais.

Quando há concentração de soluções próximas aos pontos ideais, as chances de gerar soluções filhas ainda mais próximas do ponto ideal aumenta.

Outra característica do NSGA-DO é não considerar as fronteiras durante o processo de seleção, aqui chamado de *seleção por torneio de distância*. No torneio por distância, a solução com a menor proximidade com um ponto ideal é escolhida.

Os passos do NSGA-DO são descritos no Algoritmo 4.3.

Algoritmo 4.3: NSGA-DO

```

1 Início
2    $P$ : População pai;
3    $Q$ : População filha;
4    $T$ : Tamanho fixo para  $P$  e  $Q$ ;
5    $F_j$ : Conjunto de soluções na fronteira  $j$ ;
6    $n$ : Número da geração atual;
7    $N$ : Número máximo de gerações;
8   Gerar a população inicial  $P_0$  e  $Q_0 = \{\}$ ;
9   Atribuir  $n = 0$ ;
10  Realizar a seleção, o cruzamento e a mutação em  $P_n$  para gerar a população filha  $Q_n$ ;
11  Fazer  $R_n = P_n \cup Q_n$ ;
12  Realizar a ordenação por não dominância em  $R_n$ , gerando  $F_i, i = \{1, \dots, v\}$  em  $R_n$ ;
13  Criar  $P_{n+1} = \{\}$ ;
14  Enquanto  $|P_{n+1}| + |F_i| \leq N$  Faça
15    | Copiar as soluções de  $F_i$  em  $P_{n+1}$ ;
16    |  $i = i + 1$ 
17  Fim-Enquanto
18  Para cada solução em  $F_i$  calcular a distância para os pontos ideais;
19  Ordenar  $F_i$  decrescentemente de acordo a distância para os pontos ideais;
20  Copiar as primeiras  $N - |P_{n+1}|$  soluções ordenadas de  $F_i$  para  $P_{n+1}$ ;
21  Se  $n \geq N$  Então
22    | Pare;
23  Senão
24    | Atribuir  $n = n + 1$  e voltar ao passo 10;
25  Fim-Se
26 Fim

```

4.7 Geração de Base de Conhecimento Fuzzy usando NSGA-DO

Nesta seção será apresentada a metodologia utilizada neste trabalho para geração de SFBR usando AGMO. O trabalho desenvolvido visa investigar o impacto da substituição da Distância de Multidão do NSGA-II pela proximidade com um ponto ideal na fronteira de Pareto de forma a melhorar a diversidade das soluções na fronteira de Pareto.

A metodologia utilizada é composta por duas etapas:

Seleção de regras: Nesta etapa o NSGA-DO é utilizado para selecionar regras fuzzy a partir de um *pool* de regras candidatas extraídas previamente. O objetivo é gerar uma BR capaz de representar o conhecimento existente no conjunto de exemplos e que seja ao mesmo tempo interpretável e com boa acurácia.

Otimização da Base de Dados: Nesta etapa o NSGA-DO é utilizado para otimizar os

conjuntos *fuzzy* utilizados na etapa anterior.

Com essas duas etapas, a metodologia é capaz de gerar BRs diversificadas, compactas com alta habilidade para classificação.

4.7.1 Seleção de Regras Fuzzy

Nesta etapa o NSGA-DO será responsável por selecionar regras que irão compor uma BR, a partir de um *pool* de regras. As regras do *pool* são extraídas dos conjuntos de dados a partir do método *FCA-BASED*, proposto no trabalho em Cintra (2012).

Codificação da BR

O cromossomo desta etapa, ilustrado na Figura 4.9, é um vetor de tamanho máximo previamente definido. Neste cromossomo, cada posição contém um número que serve como índice para as regras do *pool*. Este índice varia de 1 até o tamanho do *pool*. O número 0 indica a ausência de regra, o que permite diminuir o tamanho da BR tornando-a, desta forma, mais interpretável.

1	21	13	29	31	0	11	54	98	65	0	52	76	0	0
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Figura 4.9: Representação cromossômica de uma BR

A BD utilizada neste processo é uma base com três conjuntos *fuzzy* triangulares igualmente espaçados. Esta BD permanecerá fixa durante este processo.

Objetivos

Nesta etapa, o NSGA-DO terá como objetivos a maximização da Taxa de Classificação Correta (TCC) e a minimização do índice de interpretabilidade *iBR* da BR medida pela equação 4.10 dada a seguir:

$$iBR = \frac{\left(\frac{nR}{nTR}\right) + \left(\frac{nA}{nTA}\right)}{2} \quad (4.10)$$

onde:

- nR é o número de regras da BR atual;
- nTR é o tamanho máximo da BR;
- nA é o número de antecedentes da regra;

- nTA é o número total possível de antecedentes da BR. Este valor pode ser obtido multiplicando-se tBR pela quantidade de regras da BR.

Operadores genéticos

A seleção é a primeira operação a ser empregada e utiliza o método da *seleção por torneio de distância*. A segunda operação a ser realizada é o cruzamento. Neste trabalho foi utilizado o cruzamento uniforme. Os pares de cromossomos a serem cruzados e a máscara de cruzamento são escolhidos aleatoriamente. Por fim, a mutação padrão é aplicada para alterar um gene c_{ij} para um novo valor escolhido aleatoriamente dentro dos possíveis valores $0, 1, \dots, G$, sendo G o número máximo que o gene pode assumir, ou seja, o tamanho do *pool*. Para que o número de regras na BR diminua, é utilizada uma taxa $pZerar$ que determina se o operador de mutação exclui a regra de um determinado gene ou se altera a regra.

Condição de parada

O critério de parada adotado para este AGMO é o número máximo de gerações. Ao atingir esse número, o AGMO retornará um conjunto de soluções não dominadas.

4.7.2 Otimização de Base de Dados

Ao término da etapa anterior, uma das soluções não dominadas será escolhida pelo método de proximidade ao PEF. Esta BR escolhida será utilizada de maneira fixa nesta segunda etapa, que é responsável por evoluir uma população de BD para encontrar o melhor ajuste para cada conjunto de cada partição fuzzy.

Codificação da BD Os cromossomos possuem tamanho fixo e é necessário estabelecer previamente a quantidade de conjuntos que as partições terão. Os conjuntos são codificados com números reais, os quais representam os parâmetros de cada função de pertinência de cada partição.

Cada indivíduo da população representa uma BD completa. A Figura 4.10 ilustra a representação de um cromossomo que está codificando nP partições, sendo que cada partição possui nC conjuntos fuzzy.

q_{11}^1	q_{12}^1	q_{13}^1	...	q_{nC1}^1	q_{nC2}^1	q_{nC3}^1	...	q_{11}^{nP}	q_{12}^{nP}	q_{13}^{nP}	...	q_{nC1}^{nP}	q_{nC2}^{nP}	q_{nC3}^{nP}
------------	------------	------------	-----	-------------	-------------	-------------	-----	---------------	---------------	---------------	-----	----------------	----------------	----------------

Figura 4.10: Representação cromossômica de uma BD

A população inicial é gerada introduzindo um cromossomo que representa as funções de pertinência utilizadas de forma fixa na etapa 1. Os outros cromossomos são gerados aleatoriamente.

Objetivos

Nesta etapa, o NSGA-DO terá como objetivos a maximização da TCC e a maximização do índice de interpretabilidade.

O índice de interpretabilidade utilizado nesta tese é o proposto por Gacto, Alcalá e Herrera (2010), denominado *GM3M*. Este índice está focado na semântica das partições, sendo que os autores propõem algumas condições para garantir a distinguibilidade e a cobertura das partições:

- Condição de simetria: As funções de pertinência devem ser simétricas. Por definição, a simetria é garantida quando são utilizadas funções Gaussianas.
- Condição δ : Essa condição controla o ponto de intersecção de duas funções de pertinência. Esse ponto deve ser maior ou igual ao valor de δ .
- Condição γ : Essa condição controla a sobreposição no centro de cada função de pertinência. No centro de uma função de pertinência, outra função não pode ter valor maior do que γ .
- Condição ρ : Essa condição garante que o universo de discurso seja totalmente coberto pelas funções de pertinência. Para cada ponto do universo de discurso, pelo menos uma função de pertinência tem valor de pertinência maior ou igual a ρ naquele ponto.

Os valores sugeridos pelos autores são: $\delta = 0.8$, $\gamma = 0.25$ e $\rho = 0.05$.

Este índice busca manter o formato original da função de pertinência enquanto o processo de otimização dos parâmetros da função é executado. O índice é obtido pela Equação 4.11.

$$GM3M = \sqrt[3]{\delta\gamma\rho} \quad (4.11)$$

A medida δ trata do deslocamento das funções de pertinência medindo a proximidade dos pontos centrais das funções obtidas com os pontos centrais das funções originais. Quanto mais próximos eles estiverem dos pontos originais, maior será o valor de δ .

A medida γ trata da taxa de amplitude lateral. Esta métrica mede o quanto cada lateral da função mudou em relação ao formato original. Para isso, divide-se a distância entre primeiro

ponto da função e o ponto central da função, pela distância entre o ponto central e o terceiro ponto da função. Quanto mais próximas as taxas, maior a taxa de amplitude lateral.

A medida ρ trata da semelhança área. Esta métrica mede a semelhança entre a área da função otimizada e a área da função original. Quanto mais semelhante for a área, maior será o valor de ρ .

Operadores genéticos

O método de seleção aplicado nesta etapa, assim como na primeira etapa, é a *seleção por torneio de distância*. O operador de cruzamento utilizado nesta etapa é o cruzamento aritmético exemplificado na Seção 3.2.4. Por fim, a mutação é aplicada obedecendo a restrição para que os conjuntos não fiquem deformados, ou seja, com muitas sobreposições.

Condição de parada

O critério de parada adotado para este AGMO, assim como na etapa anterior, é o número máximo de gerações.

4.8 Considerações Finais

Neste capítulo foi apresentado um novo AGMO baseado no NSGA-II, denominado NSGAIIDO, que usa os pontos ideais (uniformemente distribuídos na fronteira de Pareto) como referência para guiar as soluções geradas de maneira a melhorar a diversidade das soluções na fronteira de Pareto. Uma abordagem utilizando o NSGA-DO foi proposta para a geração de bases de conhecimento *fuzzy*.

No próximo capítulo são apresentados os experimentos realizados, os resultados obtidos e a comparação entre os resultados obtidos por meio do NSGA-DO e os resultados obtidos por meio do algoritmo original NSGA-II.

Capítulo 5

EXPERIMENTOS E ANÁLISE DOS RESULTADOS

5.1 Considerações Iniciais

Para analisar o desempenho, as contribuições e as limitações do NSGA-DO, foram realizados experimentos envolvendo alguns dos conjuntos encontrados no *UCI Repository of Machine Learning Databases* (LICHMAN, 2013). Com intuito de comparação, os mesmos experimentos também foram realizados com o NSGA-II. Ambos algoritmos foram executados com as mesmas configurações.

Neste capítulo são apresentados os detalhes dos experimentos realizados, os parâmetros de configuração dos AGMOs, os resultados e a discussão sobre os resultados obtidos para a avaliação do método.

Durante a condução dos experimentos, os seguintes aspectos foram considerados:

- Dispersão das soluções na fronteira de Pareto;
- Desempenho, com relação a TCC;
- Interpretabilidade dos modelos gerados, com relação à complexidade da BR, levando em consideração o número de regras e o número de antecedentes, e com relação à semântica da BD, levando em consideração o formato e o posicionamento dos conjuntos.

5.2 Descrição dos Conjuntos de Dados

Na Tabela 5.1 são listados os conjuntos de dados do repositório UCI utilizados nos experimentos, bem como a descrição de algumas características dos conjuntos utilizados.

Tabela 5.1: Conjuntos do repositório *UCI* utilizados nos experimentos

Conjunto de dados	Exemplos	Atributos	Classes	Pool
australian	690	14	2	9998
balance	625	4	3	1834
cleveland	297	13	5	36983
contraceptive	1473	9	3	18005
dermatology	358	16	6	13646
ecoli	336	7	8	532
haberman	306	3	2	63
hayesroth	159	4	3	194
heart	269	13	2	38737
iris	150	4	3	61
newthyroid	215	5	3	75
pima	767	8	2	2438
saheart	462	9	2	7156
tae	151	5	3	275
titanic	2201	3	2	44

Os domínios de cada um dos atributos dos conjuntos de dados estão representados por valores numéricos contínuos e precisam ser granularizados em um número de conjuntos *fuzzy* que representem os valores linguísticos que cada atributo pode assumir, para permitir a generalização do conhecimento. Para cada conjunto de dados, o *pool* de regras candidatas foi extraído do trabalho (CINTRA, 2012).

5.3 Experimentos Realizados

A Tabela 5.2 apresenta a configuração dos parâmetros dos AGMOs utilizados neste trabalho. Os valores utilizados foram escolhidos com base em experimentos preliminares em que as taxas utilizadas em experimentos similares foram variadas. Para a avaliação dos resultados foi utilizada a validação cruzada para *5-folds*.

Tabela 5.2: Configuração dos parâmetros dos AGMOs

Parâmetro	BR	BD
Gerações	1000	1000
Tamanho de P	200	100
Tamanho máximo da BR	20	-
Taxa de mutação	20%	20%
Taxa de cruzamento	70%	70%
Probabilidade de zerar	50%	-

Na primeira etapa dos testes, foi analisado o comportamento do NSGA-DO durante o pro-

cesso de seleção de regras do *pool* para a construção da BR. A taxa do Índice de Distribuição (ID), definida na Seção 4.4 do Capítulo 4, foi coletada ao final de cada execução do AGMO para cada *fold*. A média dessas taxas pode ser observada na Tabela 5.3.

Tabela 5.3: Dispersão das soluções na fronteira final

Dataset	Base de Regras		Base de Dados	
	NSGAI	NSGA-DO	NSGAI	NSGA-DO
australian	0,6781	0,2448	0,6563	0,2375
balance	0,6435	0,2059	0,5419	0,0489
cleveland	0,6248	0,2066	0,5959	0,1094
contraceptive	0,5173	0,2167	0,5121	0,2959
dermatology	0	0	0	0
ecoli	0,6978	0,1901	0	0
haberman	0,5531	0,2578	0,7612	0,2812
hayesroth	0,598	0,2587	0,558	0,206
heart	0,5038	0,1858	0,559	0,2057
iris	0,5925	0,1506	0,5765	0,1624
newthyroid	0,5455	0,1669	0,2142	0,2612
pima	0,5133	0,2718	0,9917	0,3542
saheart	0,4462	0,1384	0,7434	0,2515
tae	0,4319	0,2516	0,4788	0,181
titanic	0,4293	0,1899	0,1879	0

Pode-se observar que o NSGA-DO obteve baixo valor no índice de distribuição utilizado, o que indica que a distância entre as soluções não dominadas da fronteira de Pareto está equilibrada. O mesmo não ocorre com o NSGA-II, cujo valor do ID foi maior para todos os casos, com exceção dos conjuntos *Dermatology* e *Ecoli*.

Todavia, é importante analisar se o fato do processo de busca direcionar as soluções a pontos específicos da fronteira não está influenciando de forma negativa a acurácia do modelo.

Também é necessário verificar se, de alguma maneira, o direcionamento das soluções está fazendo com que o algoritmo perca a sua habilidade de busca, deixando de encontrar soluções com melhor acurácia, ao custo de encontrar soluções com baixo valor no índice de distribuição.

Para efetuar esta análise, o desempenho dos algoritmos na etapa 1 foi medido com relação à acurácia, observando os dados da TCC e da complexidade da BR (calculada pela Equação 4.10, no Capítulo 4). Estes dados foram coletados a partir das médias das soluções centrais da fronteira final de cada *fold* e podem ser vistos na Tabela 5.4 juntamente com o número de regras da BR (#Regras) e o número total de antecedentes da BR (#Antecedentes).

Pode-se observar na Tabela 5.4 que o NSGA-DO consegue encontrar um conjunto de soluções não dominadas com valor baixo no índice de distribuição e ao mesmo tempo aumentar a

Tabela 5.4: Taxa média das soluções centrais para a BR

Dataset	TCC		Interpretabilidade		#Regras		#Antecedentes	
	NSGA-II	NSGA-DO	NSGA-II	NSGA-DO	NSGA-II	NSGA-DO	NSGA-II	NSGA-DO
australian	0,7054	0,7578	0,6257	0,5394	17,60	6,60	62,20	20,00
balance	0,6804	0,7476	0,6075	0,5678	8,00	4,20	14,20	6,60
cleveland	0,5790	0,5882	0,5985	0,5306	12,20	5,80	41,80	13,40
contraceptive	0,4657	0,4660	0,6678	0,5322	16,40	4,20	54,80	10,00
dermatology	0,3108	0,2886	0,7366	0,5238	17,20	3,00	151,40	15,40
ecoli	0,2082	0,4318	0,6429	0,5140	7,80	2,00	30,60	4,00
haberman	0,7306	0,7648	0,6333	0,5634	5,40	3,40	11,20	7,20
hayesroth	0,6219	0,7892	0,7950	0,5840	12,60	5,60	28,60	13,20
heart	0,7639	0,7926	0,6046	0,5340	14,20	6,20	52,00	17,20
iris	0,8167	0,9350	0,6513	0,5550	8,00	3,80	18,60	8,00
newthyroid	0,8302	0,8838	0,6480	0,5610	7,60	3,60	22,20	8,60
pima	0,7534	0,7632	0,6238	0,5668	13,40	7,40	36,80	18,80
saheart	0,7200	0,7324	0,6100	0,5526	12,00	7,20	32,20	18,20
tae	0,5074	0,5618	0,6730	0,5340	13,60	3,20	39,20	6,40
titanic	0,6062	0,6770	0,5983	0,5466	4,40	2,80	9,20	5,60

TCC e a interpretabilidade.

Para todos os conjuntos de dados, com exceção do conjunto *Dermatology*, o NSGA-DO obteve BRs com maior TCC e menor complexidade do que o NSGA-II logrou obter.

A mesma análise de desempenho dos algoritmos observando os dados da TCC e da interpretabilidade (calculada pela Equação 4.11) foi feita para a etapa 2. Os resultados são mostrados na Tabela 5.5.

Tabela 5.5: Taxa média das soluções centrais para a BD

Dataset	TCC		Interpretabilidade	
	NSGAII	NSGA-DO	NSGAII	NSGA-DO
australian	0,7472	0,7080	0,9902	0,9858
balance	0,6632	0,6844	0,9042	0,9248
cleveland	0,5714	0,5678	0,9794	0,9792
contraceptive	0,4622	0,4630	0,9800	0,9776
dermatology	0,3110	0,3110	1,0000	1,0000
ecoli	0,4318	0,4318	1,0000	1,0000
haberman	0,7716	0,7688	0,9198	0,9650
hayesroth	0,7422	0,7298	0,9860	0,9904
heart	0,7712	0,7362	0,9856	0,9916
iris	0,8916	0,8918	0,9022	0,9524
newthyroid	0,6558	0,7348	0,9654	0,9684
pima	0,7650	0,7606	0,8276	0,8452
saheart	0,7212	0,7206	0,9122	0,9230
tae	0,5258	0,5356	0,9558	0,9760
titanic	0,7018	0,6770	0,9896	1,0000

A Tabela 5.5 mostra que, para a etapa 2, apesar do NSGA-II apresentar TCC maior que o do NSGA-DO para 8 conjuntos de dados, os valores estão próximos. Entretanto, para a interpretabilidade, o NSGA-DO teve desempenho superior ao NSGA-II para 10 conjuntos de

dados e empatando em outros 2 conjuntos.

Considerando que os dados que compõem as tabelas 5.4 e 5.5 foram obtidos a partir da média dos valores obtidos ao final de cada *fold*, analisou-se também a possibilidade de existir alguma fronteira, de algum *fold*, em que algumas soluções do NSGA-II que estivessem dominando as do NSGA-DO. Em caso afirmativo, isto significaria que há soluções boas que o NSGA-DO não foi capaz de encontrar.

Para ilustrar esta situação, considere-se a Figura 5.1, em que o valor da função objetivo no eixo x deve ser minimizado e o valor da função objetivo no eixo y deve ser maximizado.

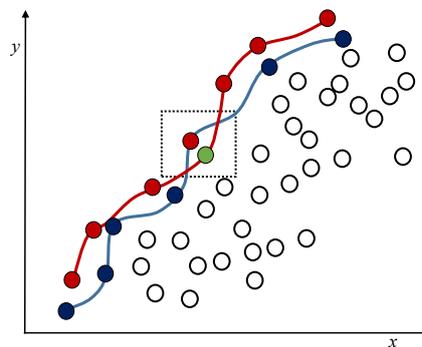


Figura 5.1: Problema de dominância

Nesta figura, tem-se duas fronteiras. Os círculos vermelhos representam as soluções não dominadas que formam a fronteira de Pareto. Pode-se observar que, em geral, as soluções da fronteira azul não apresentam melhor desempenho do que as soluções da fronteira vermelha. Isso faz com que a maioria das soluções da fronteira vermelha venha a dominar as soluções da fronteira azul. Supondo que a fronteira azul é a fronteira de Pareto gerada pelo algoritmo A em um determinado *fold* e que a fronteira vermelha é a fronteira de Pareto gerada pelo algoritmo B no mesmo *fold*, a tabela de comparação recairá sobre as soluções indicadas dentro do quadrado tracejado nesta figura. Observa-se, todavia, que essas duas soluções não representam o real desempenho dos algoritmos em questão, trazendo a falsa ideia de que as soluções do algoritmo A são melhores do que as do algoritmo B .

Neste exemplo o algoritmo B irá ter um desempenho inferior ao do algoritmo A quando usado na comparação somente a solução central de cada fronteira, sendo que todas as outras soluções do algoritmo B dominam as outras soluções do algoritmo A .

Com o intuito de analisar esta situação, para cada conjunto de dados, foram plotadas todas as soluções não dominadas da última geração dos dois AGMOs de todos os *folders* no mesmo gráfico conforme mostram as figuras 5.2, 5.3, 5.4 e 5.5.

Nas figuras 5.2 e 5.3 pode-se observar que as soluções que compõem as fronteiras do

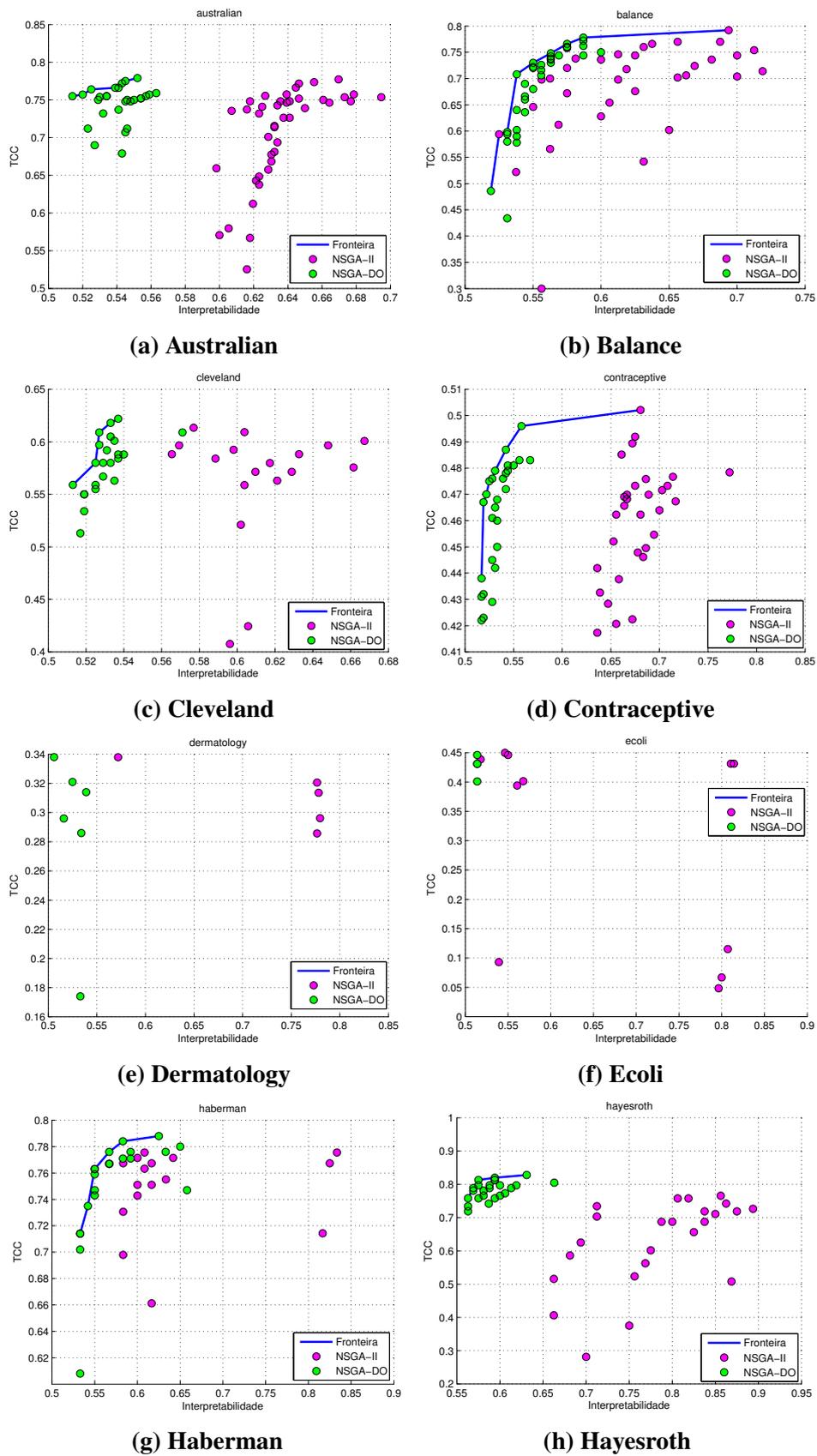


Figura 5.2: Dominância entre o NSGA-II e o NSGA-DO para a seleção de regras - Parte I.

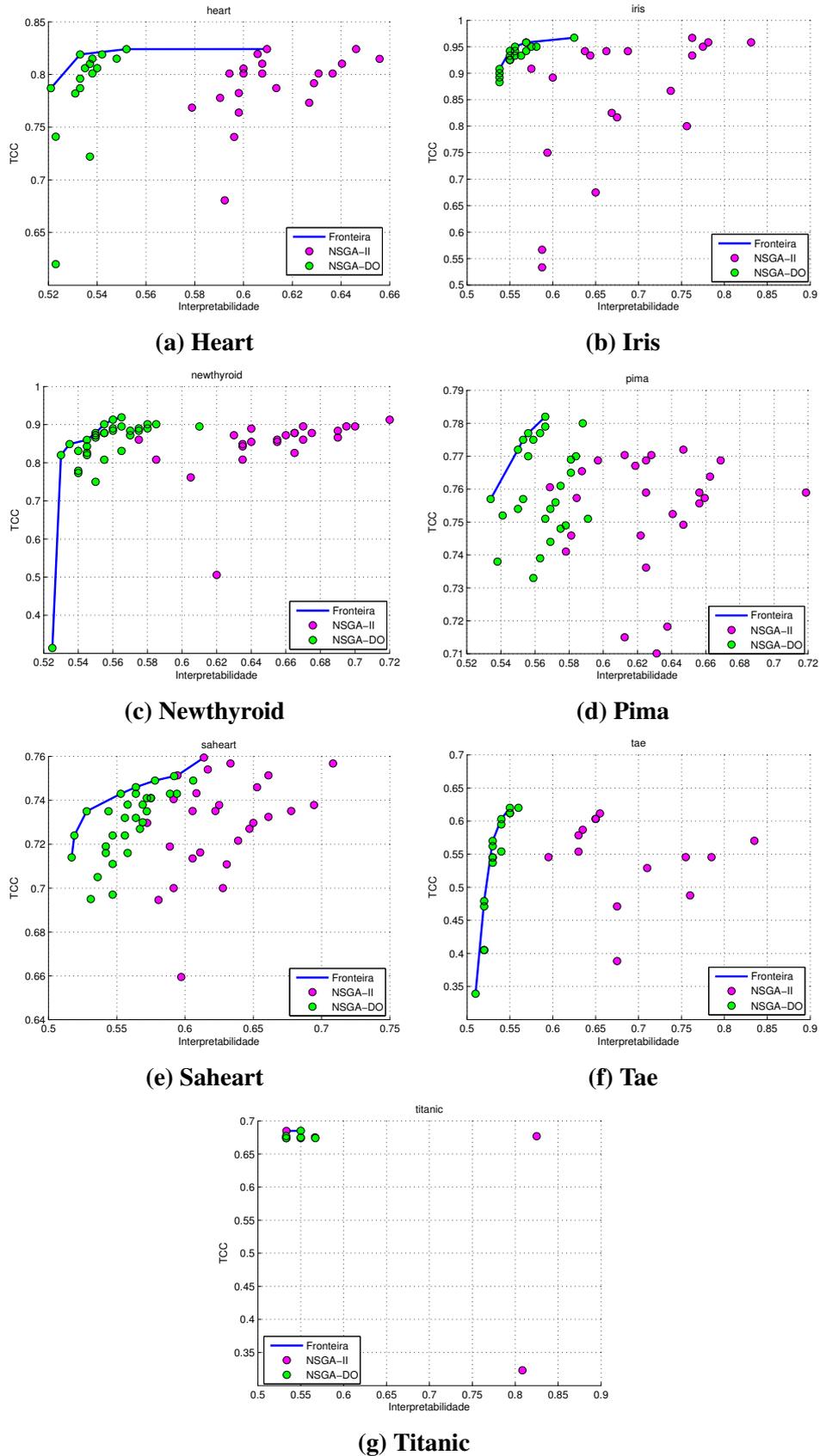


Figura 5.3: Dominância entre o NSGA-II e o NSGA-DO para a seleção de regras - Parte II.

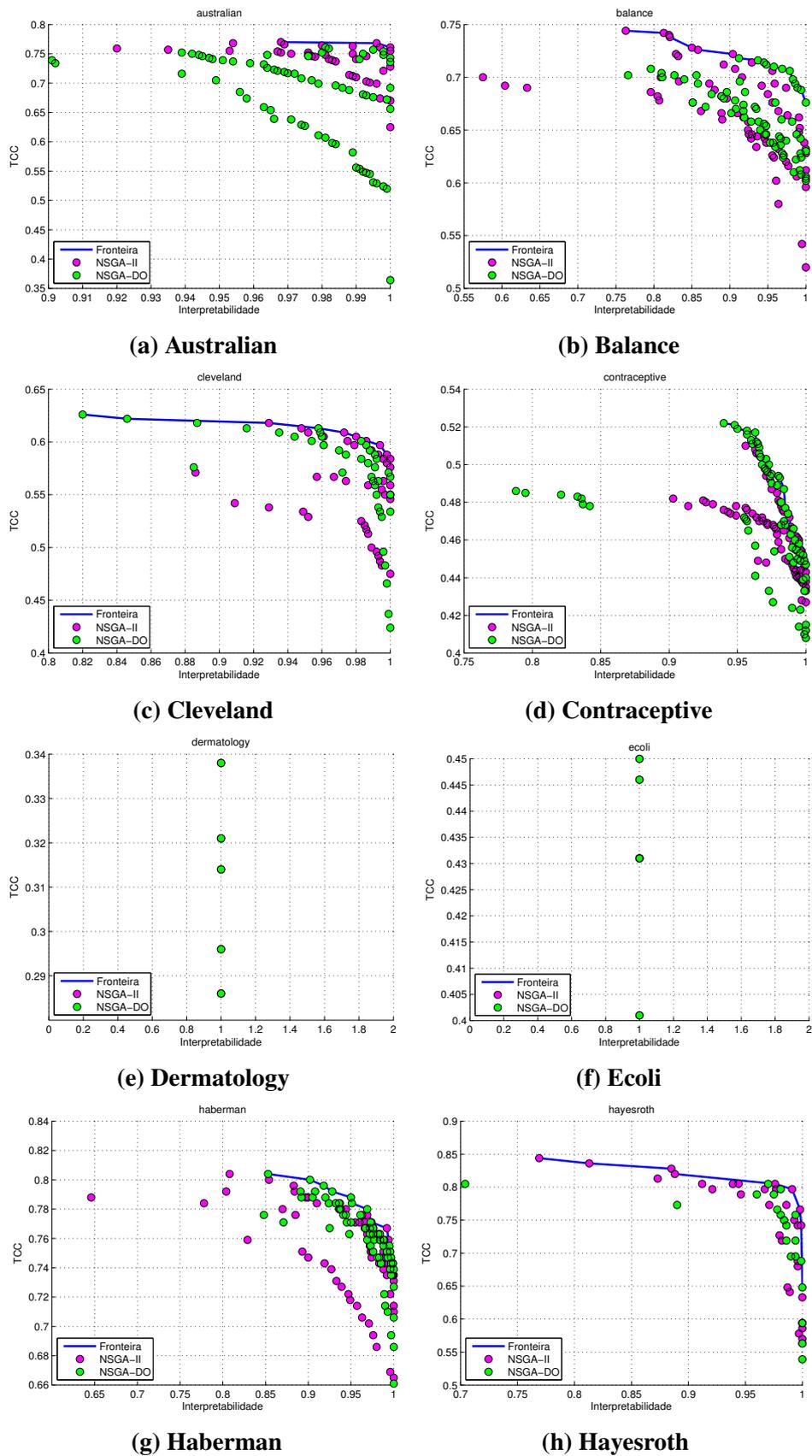


Figura 5.4: Dominância entre o NSGA-II e o NSGA-DO para a otimização da BD - Parte I.

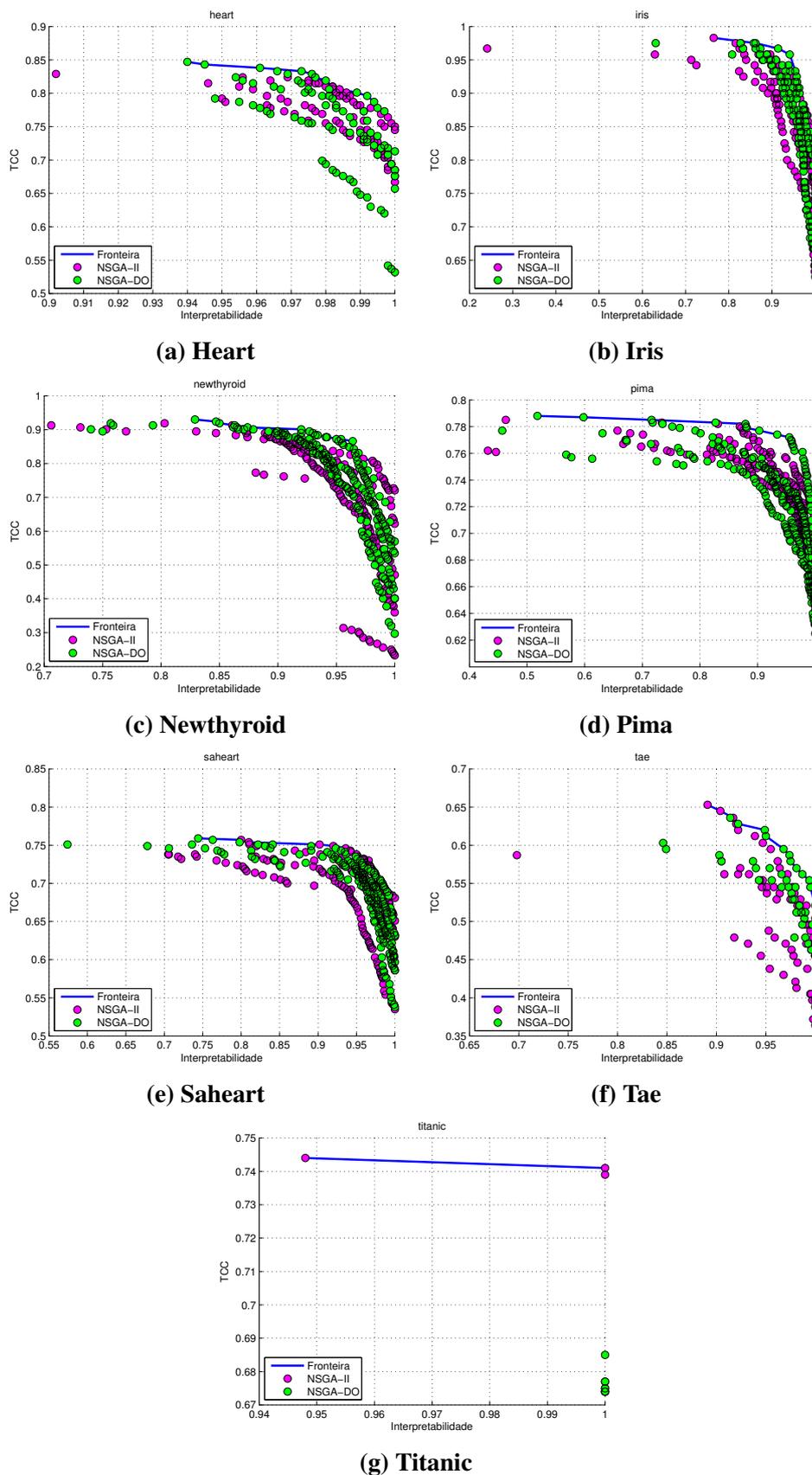


Figura 5.5: Dominância entre o NSGA-II e o NSGA-DO para a otimização da BD - Parte II.

NSGA-DO possuem distância significativa com relação às que compõem as fronteiras do NSGA-II, e estão mais próximas da fronteira real de Pareto, eliminando portanto a hipótese de que o problema ilustrado na Figura 5.1 pudesse estar acontecendo com o algoritmo proposto durante a etapa de seleção de regras.

De acordo com as figuras 5.2b, 5.2d, 5.3a, 5.3e e 5.3g, pode-se observar que os conjuntos *Balance*, *Contraceptive*, *Heart*, *Saheart* e *Titanic* foram os únicos conjuntos em que soluções do NSGA-II pertencem à fronteira de Pareto. Entretanto, o número de soluções encontradas pelo NSGA-II que compõem a fronteira de Pareto é muito menor do que o número de soluções encontradas pelo NSGA-DO que compõem a fronteira de Pareto.

Nas figuras 5.2a, 5.2d, 5.2e, 5.2f, 5.3a e 5.3f pode-se perceber que do ponto de vista da interpretabilidade, a pior solução do NSGA-DO, isto é, a de mais baixa interpretabilidade, possui interpretabilidade maior do que a melhor solução do NSGA-II no mesmo objetivo. As figuras 5.2h e 5.3b indicam que o mesmo pode ocorrer do ponto de vista da TCC.

Na Figura 5.4 pode-se observar que há muita sobreposição entre as soluções, o que impede afirmar que, visualmente, as fronteiras dos dois algoritmos possuem distância significativa entre si. Desta forma, não é possível verificar se ocorre o problema ilustrado pela Figura 5.1.

Para avaliar de maneira mais segura o grau de dominância entre as soluções dos dois algoritmos, foi calculado quantas das soluções de cada algoritmo compõem a fronteira de Pareto. Esses dados podem ser observados na Tabela 5.6.

Tabela 5.6: Número de soluções na fronteira ótimo de Pareto

Dataset	Base de Regras				Base de Dados			
	NSGAI	NSGAI	NSGA-DO	NSGA-DO	NSGAI	NSGAI	NSGA-DO	NSGA-DO
	Qtd.	(%)	Qtd.	(%)	Qtd.	(%)	Qtd.	(%)
australian	0	0%	7	100%	3	100%	0	0%
balance	2	22%	7	78%	7	37%	12	63%
cleveland	0	0%	5	100%	7	70%	3	30%
contraceptive	1	11%	8	89%	9	30%	21	70%
dermatology	0	0%	1	100%	1	50%	1	50%
ecoli	0	0%	1	100%	1	50%	1	50%
haberman	0	0%	8	100%	3	19%	13	81%
hayesroth	0	0%	5	100%	8	89%	1	11%
heart	1	25%	3	75%	6	33%	12	67%
iris	0	0%	6	100%	7	35%	13	65%
newthyroid	0	0%	8	100%	12	46%	14	54%
pima	0	0%	5	100%	8	22%	28	78%
saheart	2	22%	7	78%	25	76%	8	24%
tae	0	0%	5	100%	3	21%	11	79%
titanic	1	33%	2	67%	2	100%	0	0%

Pode-se observar que há um alto grau de dominância das soluções do NSGA-DO sobre as soluções do NSGA-II para o processo de seleção de regras.

Os dados apresentados na Tabela 5.6 comprovam que o grau de dominância das soluções do NSGA-DO sobre as soluções do NSGA-II é alto para a primeira etapa de seleção de regras. Para o processo de otimização da BD, pode-se observar que a maior parte das soluções que formam a fronteira de Pareto provém do NSGA-DO para a maioria dos conjuntos de dados, evidenciando o alto grau de dominância das soluções encontradas pelo NSGA-DO.

Para cada um dos conjuntos de dados, foi feita também a análise da dominância entre todas as soluções não dominadas da última geração dos dois AGMOs de todos os *folds* da etapa de seleção de regras. O resultado desta análise pode ser visto nas tabelas 5.7 e 5.8.

Tabela 5.7: Dominância do NSGA-II sobre o NSGA-DO na etapa de seleção de regras

Dataset	100% a 71%		70% a 41%		40% a 21%		20% a 5%	
	# Sol.	%	# Sol.	%	# Sol.	%	# Sol.	%
Australian	0	0,00%	0	0,00%	0	0,00%	0	0,00%
Balance	0	0,00%	0	0,00%	0	0,00%	1	2,94%
Cleveland	0	0,00%	0	0,00%	0	0,00%	0	0,00%
Contraceptive	0	0,00%	0	0,00%	0	0,00%	0	0,00%
Dermatology	0	0,00%	0	0,00%	0	0,00%	0	0,00%
Ecoli	0	0,00%	0	0,00%	0	0,00%	0	0,00%
Haberman	0	0,00%	0	0,00%	0	0,00%	0	0,00%
Hayesroth	0	0,00%	0	0,00%	0	0,00%	0	0,00%
Heart	0	0,00%	0	0,00%	0	0,00%	0	0,00%
Iris	0	0,00%	0	0,00%	0	0,00%	0	0,00%
Newthyroid	0	0,00%	0	0,00%	0	0,00%	0	0,00%
Pima	0	0,00%	0	0,00%	1	4,00%	0	0,00%
Saheart	0	0,00%	0	0,00%	0	0,00%	0	0,00%
Tae	0	0,00%	0	0,00%	0	0,00%	0	0,00%
Titanic	0	0,00%	0	0,00%	1	16,67%	3	50,00%

Na Tabela 5.7 foram separadas as soluções do NSGA-II em quatro grupos:

- Quantidade de soluções do NSGA-II que dominam entre 100% e 71% das soluções do NSGA-DO;
- Quantidade de soluções do NSGA-II que dominam entre 70% e 41% das soluções do NSGA-DO;
- Quantidade de soluções do NSGA-II que dominam entre 40% e 21% das soluções do NSGA-DO;

Tabela 5.8: Dominância do NSGA-DO sobre o NSGA-II na etapa de seleção de regras

Dataset	100% a 71%		70% a 41%		40% a 21%		20% a 5%	
	# Sol.	%	# Sol.	%	# Sol.	%	# Sol.	%
Australian	19	67,86%	4	14,29%	5	17,86%	0	0,00%
Balance	0	0,00%	22	61,11%	4	11,11%	8	22,22%
Cleveland	7	30,43%	7	30,43%	4	17,39%	5	21,74%
Contraceptive	12	42,86%	4	14,29%	5	17,86%	6	21,43%
Dermatology	2	33,33%	1	16,67%	0	0,00%	2	33,33%
Ecoli	2	40,00%	3	60,00%	0	0,00%	0	0,00%
Haberman	4	18,18%	7	31,82%	5	22,73%	5	22,73%
Hayesroth	24	100,00%	0	0,00%	0	0,00%	0	0,00%
Heart	8	50,00%	2	12,50%	3	18,75%	1	6,25%
Iris	9	47,37%	10	52,63%	0	0,00%	0	0,00%
Newthyroid	15	45,45%	4	12,12%	2	6,06%	5	15,15%
Pima	11	40,74%	2	7,40%	10	37,04%	4	14,81%
Saheart	2	6,45%	14	45,16%	8	25,81%	4	19,20%
Tae	6	35,29%	3	17,65%	3	17,65%	4	23,53%
Titanic	1	20,00%	1	20,00%	1	20,00%	2	40,00%

- Quantidade de soluções do NSGA-II que dominam entre 20% e 5% das soluções do NSGA-DO.

Pela análise desta tabela vê-se que o grau de dominância das soluções do NSGA-II sobre as soluções do NSGA-DO, para a etapa de seleção de regras, é praticamente nulo, sendo que apenas dois conjuntos de dados (*Pima* e *Titanic*) possuem uma única solução que domina somente entre 40% e 21% das soluções do NSGA-DO. Ainda por esta tabela pode-se observar que o conjunto *Balance* possui uma solução que domina entre 20% e 5% das soluções do NSGA-DO e o conjunto *Titanic* possui três soluções que dominam soluções do NSGA-DO também entre 20% e 5%.

A Tabela 5.8 faz a separação das soluções também em quatro grupos, porém considerando agora a quantidade de soluções do NSGA-DO que dominam as soluções do NSGA-II em cada grupo.

Para catorze dos quinze conjuntos de dados utilizados, o NSGA-DO gerou soluções que dominam entre 100% e 71 % das soluções do NSGA-II. Para o conjunto Hayesroth, 100% das soluções geradas pelo NSGA-DO dominam entre 100% e 71% das soluções geradas pelo NSGA-II.

Utilizando as tabelas 5.7 e 5.8 verificou-se a quantidade de conjuntos de dados em que o NSGA-II tem mais soluções que dominam o NSGA-DO, a quantidade de conjuntos de dados em que o NSGA-DO tem mais soluções que dominam o NSGA-II e a quantidade de conjuntos de

dados em que o número de soluções de um algoritmo dominam as soluções do outro algoritmo é igual. Esses valores podem ser observados na Tabela 5.9.

Tabela 5.9: Maior dominância por conjunto de dados na etapa 1

	NSGA-II > NSGA-DO	NSGA-DO > NSGA-II	NSGA-II = NSGA-DO
100% a 71%	0	14	1
70% a 41%	0	14	1
40% a 21%	0	10	5
20% a 5%	1	10	4

A Tabela 5.9 mostra com clareza em quantos conjuntos o NSGA-DO possui uma quantidade maior de soluções que dominam as soluções do NSGA-II para cada intervalo definido.

O único caso de empate para o intervalo entre 100% e 71% foi para o conjunto de dados Balance. Para este conjunto, a dominância do NSGA-DO sobre o NSGA-II ficou concentrada no intervalo entre 70% e 40%.

O empate ocorrido para o intervalo entre 70% e 40% foi para o conjunto de dados Hayes-roth. Para este conjunto, todas as soluções não dominadas encontradas pelo NSGA-DO dominaram entre 100% e 71% das soluções do NSGA-II.

Os outros casos de empate são justificados pela concentração das soluções não dominadas encontradas pelo NSGA-DO em intervalos de dominância maiores.

O mesmo procedimento de comparação foi feito para a etapa de otimização da BD. Podemos verificar os resultados nas Tabelas 5.10 e 5.11.

Para facilitar a análise dos resultados destas tabelas, verificou-se também a quantidade de conjuntos de dados em que o NSGA-II tem mais soluções que dominam o NSGA-DO, a quantidade de conjuntos de dados em que o NSGA-DO tem mais soluções que dominam o NSGA-II e a quantidade de conjuntos de dados em que o número de soluções de um algoritmo domina as soluções do outro algoritmo é igual. Esses valores podem ser observados na Tabela 5.12.

Pode-se observar nesta tabela que para o intervalo entre 100% e 71% há empate para a maior parte dos conjuntos de dados. Para o intervalo entre 70% e 41%, o NSGA-II obteve dominância em um número um pouco maior de conjunto de dados (6 para o NSGA-II contra 4 para o NSGA-DO). Já para o intervalo entre 40% e 21%, há um empate entre as dominâncias dos dois algoritmos. Por fim, no intervalo entre 20% e 5%, o NSGA-DO possui uma quantidade maior de soluções que dominam o NSGA-II.

Com o intuito de analisar a significância estatística das diferenças entre o algoritmo NSGA-

Tabela 5.10: Dominância do NSGA-II sobre o NSGA-DO na etapa de otimização da BD

Dataset	100% a 71%		70% a 41%		40% a 21%		20% a 5%	
	# Sol.	%	# Sol.	%	# Sol.	%	# Sol.	%
Australian	4	10,53%	14	36,84%	15	39,47%	2	5,26%
Balance	0	0,00%	2	3,03%	11	16,67%	14	21,21%
Cleveland	0	0,00%	4	9,75%	7	17,07%	12	29,27%
Contraceptive	0	0,00%	0	0,00%	0	0,00%	57	74,03%
Dermatology	1	20,00%	1	20,00%	1	20,00%	1	20,00%
Ecoli	1	20,00%	1	20,00%	0	0,00%	2	40,00%
Haberman	0	0,00%	0	0,00%	2	2,99%	24	35,82%
Hayesroth	0	0,00%	2	6,25%	4	12,50%	10	31,25%
Heart	0	0,00%	4	5,06%	20	25,32%	38	48,10%
Iris	0	0,00%	0	0,00%	9	6,77%	29	21,80%
Newthyroid	0	0,00%	2	1,21%	17	10,24%	20	12,05%
Pima	0	0,00%	0	0,00%	25	11,63%	130	60,47%
Saheart	0	0,00%	1	0,60%	33	20,00%	35	21,21%
Tae	0	0,00%	0	0,00%	1	1,61%	20	32,26%
Titanic	2	33,33%	1	16,67%	1	16,67%	0	0,00%

Tabela 5.11: Dominância do NSGA-DO sobre o NSGA-II na etapa 2 (Otimização BD)

Dataset	100% a 71%		70% a 41%		40% a 21%		20% a 5%	
	# Sol.	%	# Sol.	%	# Sol.	%	# Sol.	%
Australian	0	0,00%	4	5,97%	4	5,97%	3	4,48%
Balance	0	0,00%	11	16,18%	6	8,82%	38	55,88%
Cleveland	0	0,00%	2	4,87%	19	46,34%	11	26,83%
Contraceptive	0	0,00%	1	1,38%	19	26,39%	23	31,94%
Dermatology	1	20,00%	1	20,00%	1	20,00%	1	20,00%
Ecoli	1	20,00%	1	20,00%	0	0,00%	2	40,00%
Haberman	0	0,00%	0	0,00%	24	36,36%	33	50,00%
Hayesroth	0	0,00%	0	0,00%	1	4,76%	12	57,14%
Heart	0	0,00%	2	2,38%	5	5,95%	29	34,52%
Iris	0	0,00%	0	0,00%	29	22,83%	58	45,67%
Newthyroid	0	0,00%	0	0,00%	35	17,68%	114	57,58%
Pima	0	0,00%	5	2,16%	17	7,32%	59	25,43%
Saheart	0	0,00%	0	0,00%	23	12,04%	141	73,82%
Tae	0	0,00%	5	10,87%	17	36,96%	13	28,26%
Titanic	0	0,00%	1	20,00%	1	20,00%	1	20,00%

Tabela 5.12: Maior dominância por conjunto de dados na etapa 2

	NSGA-II > NSGA-DO	NSGA-DO > NSGA-II	NSGA-II = NSGA-DO
100% a 71%	2	0	13
70% a 41%	6	4	5
40% a 21%	6	6	3
20% a 5%	5	8	2

II e o algoritmo NSGA-DO, foi aplicado o teste de *Wilcoxon* para todas as tabelas de comparação.

Na Tabela 5.13 são apresentados os resultados dos testes estatísticos.

Tabela 5.13: Teste Wilcoxon

NSGA-II vs NSGA-DO		R_+	R_-	H0 ($\alpha = 0,05$)	Valor p
BR	Acurácia	5	115	Rejeitada	0,0006
	Interpretabilidade	120	0	Rejeitada	0,0001
	Dispersão de soluções	105	0	Rejeitada	0,0001
	# soluções na fronteira	0	120	Rejeitada	0,0007
BD	Acurácia	58	33	Aceita	0,4143
	Interpretabilidade	7	84	Rejeitada	0,0046
	Dispersão de soluções	90	1	Aceita	0,0005
	# soluções na fronteira	28,5	62,5	Rejeitada	0,2486

Supondo como hipótese nula h_0 que *não há diferença significativa entre as soluções obtidas pelo NSGA-II e as soluções obtidas pelo NSGA-DO*, pode-se observar que tanto para a etapa de seleção de regras, quanto para a etapa de otimização da BD, o NSGA-DO mostrou-se estatisticamente mais significativo que o NSGA-II.

Podemos ver que para os dois estágios, Seleção de Regras e Otimização da BD, o NSGA-DO foi estatisticamente mais significativo que o NSGA-II. A acurácia do sistema na etapa de otimização da BD não foi significativamente diferente. Isso demonstra que o NSGA-DO obtém maior interpretabilidade do que o NSGA-II sem perder significativamente a precisão. O número de soluções encontradas na fronteira de Pareto não foi significativamente diferente. No entanto, o baixo valor do Índice de Distribuição, combinado com uma boa acurácia e maior interpretabilidade tornam a NSGA-DO uma proposta promissora.

5.4 Considerações Finais

Neste capítulo foram apresentados os experimentos realizados com o NSGA-DO e com o NSGA-II aplicando-os na seleção de regras *fuzzy* a partir de um *pool* de regras e, em seguida, otimizando a BD.

Os experimentos realizados mostraram que o NSGA-DO foi capaz de encontrar soluções distribuídas ao longo da fronteira de Pareto de forma mais uniforme do que o NSGA-II aumentando, assim, o equilíbrio entre as soluções encontradas com relação a TCC e a interpretabilidade ao longo da fronteira de Pareto.

Além de encontrar fronteiras com soluções melhor distribuídas, o NSGA-DO foi capaz de encontrar soluções centrais com TCC e interpretabilidade maiores do que o NSGA-II.

Uma análise sobre a dominância das soluções foi feita com o objetivo de verificar a qualidade de todas as soluções encontradas na fronteira de Pareto. Esta análise mostrou que as soluções encontradas pelo NSGA-DO, além de apresentarem uma distribuição mais uniforme, também dominam a maior parte das soluções encontradas pelo NSGA-II.

A partir dos resultados obtidos com os experimentos realizados, o próximo capítulo descreve as conclusões da tese, discutindo as principais contribuições e limitações da abordagem desenvolvida. Além disso, serão apresentadas as possibilidades de expansão da pesquisa com trabalhos futuros.

Capítulo 6

CONCLUSÕES E TRABALHOS FUTUROS

6.1 Considerações Iniciais

Neste trabalho foi investigada a distribuição das soluções não dominadas na fronteira de Pareto e temas relacionados a este problema, tais como o comprimento da fronteira de Pareto, a identificação dos pontos ideais na fronteira de Pareto, a distância entre estes pontos ideais e as soluções encontradas por um AGMO no contexto de classificação *fuzzy*.

Neste capítulo final, listaremos as contribuições deste trabalho e os possíveis trabalhos futuros a serem desenvolvidos.

6.2 Contribuições

Uma das principais contribuições deste trabalho é a introdução da ideia de orientar as soluções de um AGMO com base na distância das soluções encontradas por este AGMO e os pontos ideais da fronteira de Pareto, ou seja, os pontos que representam a melhor distribuição das soluções ao longo da fronteira de Pareto. Neste sentido, foi proposto o NSGA-DO, que substituiu a distância de multidão usada pelo algoritmo NSGA-II pela distância mínima entre as soluções encontradas e os pontos ideais da fronteira de Pareto.

Foi proposto neste trabalho o uso do algoritmo NSGA-DO para geração de bases de conhecimento *fuzzy* em uma metodologia dividida em duas etapas: uma etapa responsável pela seleção de regras e uma outra responsável pela otimização da BD. Em ambas, o NSGA-DO apresentou melhor desempenho do que o NSGA-II quanto à distribuição das soluções, à acurácia e à interpretabilidade.

Desta maneira, cumpriu-se o principal objetivo deste trabalho de doutorado:

Criar um método que permita a um AGMO encontrar soluções com alto grau de diversidade na fronteira de Pareto. O AGMO será usado na geração automática de um SCF, mantendo o equilíbrio entre a acuidade e a interpretabilidade.

Neste trabalho mostrou-se que, a partir do cálculo das funções que representam a fronteira de Pareto, podem ser encontrados os pontos ideais desta fronteira. Estes pontos são, então, utilizados como referência para o NSGA-DO selecionar as soluções para as próximas gerações.

Os pontos ideais foram explorados, também, para analisar a distribuição das soluções ao longo da fronteira de Pareto. Com este intuito, foi proposto um índice que mede quão bem distribuídas estão as soluções ao longo da fronteira de Pareto. Este índice reflete com precisão a distribuição das soluções ao longo da fronteira, pois considera os pontos de inflexão que podem ser encontrados na fronteira de Pareto.

Outra contribuição deste trabalho foi a criação de um método para a escolha de uma solução na fronteira de Pareto. Este método, baseado nos pontos ideais da fronteira, é capaz de selecionar a solução que possui o maior balanceamento entre os objetivos a serem otimizados, provando desta forma a hipótese que dirigiu este trabalho:

Considerar a distância entre as soluções presentes na fronteira de Pareto, durante o processo evolutivo de um AGMO, como uma medida de avaliação da qualidade do balanceamento entre acuidade e interpretabilidade, fará com que este AGMO convirja para uma fronteira de Pareto cujas soluções estejam distribuídas de uma maneira próxima a uma distribuição uniforme ao longo da fronteira de Pareto.

A Tabela 6.1 sumariza os métodos propostos neste trabalho distribuídos de acordo com a tarefa que eles tratam.

Tabela 6.1: Métodos propostos neste trabalho

Tarefa	Proposta
Determinar o espaçamento ideal entre as soluções	Capítulo 4 Seção 4.3
Índice de distribuição das soluções	Capítulo 4 Seção 4.4
Escolha da solução final	Capítulo 4 Seção 4.5
Manutenção da diversidade das soluções na otimização genética multiobjetivo	Capítulo 4 Seção 4.6

6.3 Limitações e Trabalhos Futuros

A partir deste trabalho surgem novas oportunidades de pesquisa, tanto no sentido de melhorar os métodos propostos, quanto na elaboração de novas ideias.

Embora o uso dos pontos ideais como base para as propostas apresentadas neste trabalho tenham apresentado bons resultados, o método usado para encontrar estes pontos ideais está concentrado no espaço de busca bidimensional, o que traz uma limitação para o método. Para que este método possa contribuir com classes de problemas de otimização em um espaço de busca com mais de duas dimensões, é preciso estendê-lo para que possa lidar com estes problemas.

O método proposto neste trabalho pode ser aplicado a outros conjuntos de dados e comparado com outros métodos encontrados na literatura para verificar se a tendência de obtenção de melhores resultados com o NSGA-DO se mantém.

Não obstante, o conceito de distância com relação aos pontos ideais pode ser introduzido no operador de cruzamento. Uma possível abordagem é escolher as n soluções mais próximas do ponto ideal I_k com o intuito de gerar soluções próximas a este ponto ideal. Para evitar a convergência prematura, os outros indivíduos não seriam gerados por meio desta abordagem.

Por fim, a ideia de seleção dirigida por distância pode, também, ser estendida a outros algoritmos genéticos multiobjetivos.

REFERÊNCIAS

ABADEH, M.; HABIBI, J.; LUCAS, C. Intrusion detection using a fuzzy genetics-based learning algorithm. *Journal of Network and Computer Applications*, v. 30, p. 414–428, 2007. ISSN 10848045. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S1084804505000305>>.

ABRAHAM, A.; JAIN, L.; GOLDBERG, R. *Evolutionary Multiobjective Optimization*. London: Springer-Verlag, 2005. 1–9 p. (Advanced Information and Knowledge Processing).

ADRA, S. F.; FLEMING, P. J. Diversity Management in Evolutionary Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation*, v. 15, n. 2, p. 183–195, abr. 2011. ISSN 1089-778X.

AGUILAR-LASSERRE, A. et al. Enhanced genetic algorithm-based fuzzy multiobjective strategy to multiproduct batch plant design. *Applied Soft Computing*, v. 9, n. 4, p. 1321–1330, set. 2009.

ALCALÁ-FDEZ, J. et al. Increasing fuzzy rules cooperation based on evolutionary adaptive inference systems. *International Journal of Intelligent Systems*, v. 22, n. 9, p. 1035–1064, set. 2007. ISSN 08848173. Disponível em: <<http://doi.wiley.com/10.1002/int.20237>>.

ALCALÁ, R. et al. A multiobjective evolutionary approach to concurrently learn rule and data bases of linguistic fuzzy-rule-based systems. *IEEE Transactions on Fuzzy Systems*, v. 17, p. 1106–1122, 2009. ISSN 10636706.

ALCALÁ, R. et al. Generating single granularity-based fuzzy classification rules for multiobjective genetic fuzzy rule selection. In: *2009 IEEE International Conference on Fuzzy Systems*. [S.l.]: IEEE, 2009. p. 1718–1723.

ALONSO, J. M.; MAGDALENA, L.; GONZÁLEZ-RODRÍGUEZ, G. Looking for a good fuzzy system interpretability index: An experimental approach. *International Journal of Approximate Reasoning*, v. 51, n. 1, p. 115–134, dez. 2009.

ANTONELLI, M. et al. A Multi-Objective Genetic Approach to Concurrently Learn Partition Granularity and Rule Bases of Mamdani Fuzzy Systems. In: *Hybrid Intelligent Systems, 2008. HIS '08. Eighth International Conference on*. [S.l.: s.n.], 2008. p. 278–283.

ANTONELLI, M.; DUCANGE, P.; MARCELLONI, F. A fast and efficient multi-objective evolutionary learning scheme for fuzzy rule-based classifiers. *Information Sciences*, v. 283, p. 36–54, 2014. ISSN 00200255.

- AZEVEDO, C. R. B.; ARAUJO, A. F. R. Correlation between diversity and hypervolume in evolutionary multiobjective optimization. In: *2011 IEEE Congress of Evolutionary Computation (CEC)*. [S.l.]: IEEE, 2011. p. 2743–2750. ISBN 978-1-4244-7834-7. ISSN Pending.
- B. Filipic, D. J. A genetic algorithm to support learning fuzzy control rules from examples. *Genetic Algorithms and Soft Computing*, 1996.
- BÄCK, T.; SCHWEFEL, H.-P. An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary Computation*, v. 1, n. 1, p. 1–23, mar. 1993.
- BANZHAF, W. et al. *Genetic Programming: An Introduction On the Automatic Evolution of Computer Programs and Its Applications*. [S.l.]: Morgan Kaufmann Publishers Inc., 1998. 470 p.
- BODENHOFER, U.; BAUER, P. A formal model of interpretability of linguistic variables. *Physica-Verlag*, v. 128, n. 2, p. 524–454, 2003.
- CANNONE, R.; ALONSO, J. M.; MAGDALENA, L. Multi-objective design of highly interpretable fuzzy rule-based classifiers with semantic cointension. In: *Genetic and Evolutionary Fuzzy Systems (GEFS), 2011 IEEE 5th International Workshop on*. [S.l.: s.n.], 2011. p. 1–8.
- CÁRDENAS, E. H.; CAMARGO, H. A. Multiobjective genetic generation of fuzzy classifiers using the iterative rule learning. In: *IEEE International Conference on Fuzzy Systems*. [S.l.: s.n.], 2012. ISBN 9781467315067. ISSN 10987584.
- CASILLAS, J.; CARSE, B.; BULL, L. Fuzzy-XCS: A Michigan genetic fuzzy system. *Fuzzy Systems, IEEE Transactions on*, IEEE, v. 15, n. 4, p. 536–550, 2007.
- CHAMBERS, L. *Genetic algorithms*. [S.l.]: Chapman and Hall/CRC, Boca Raton, 2001.
- CHI, Z.; YAN, H.; PHAM, T. fuzzy algorithms with applications to image processing and pattern recognition. *World Scientific*, 1996.
- CINTRA, M. E. *Genetic generation of fuzzy knowledge bases: new perspectives*. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2012. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-16072012-144620/>>.
- COELLO, C. Evolutionary multi-objective optimization: a historical view of the field. *IEEE Computational Intelligence Magazine*, v. 1, n. 1, p. 28–36, fev. 2006.
- CORDÓN, O. A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems. 2011. 894–913 p.
- CORDÓN, O. et al. Genetic Fuzzy Systems: What’s Next? An Introduction to the Special Section. *IEEE Transactions on Fuzzy Systems*, v. 15, n. 4, p. 533–535, ago. 2007.
- CORDÓN, O. et al. Ten years of genetic fuzzy systems: current framework and new trends. *Fuzzy Sets and Systems*, v. 141, n. 1, p. 5–31, jan. 2004.

- CORDON, O. et al. A multiobjective genetic algorithm for feature selection and granularity learning in fuzzy-rule based classification systems. *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569)*, v. 3, 2001.
- CORNE, D. et al. PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO2001)*. [S.l.: s.n.], 2001. p. 283–290.
- CORNE, D.; KNOWLES, J.; OATES, M. The Pareto Envelope-Based Selection Algorithm for Multiobjective Optimization. In: SCHOENAUER, M. et al. (Ed.). *Parallel Problem Solving from Nature-PPSN VI*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, (Lecture Notes in Computer Science, v. 1917). p. 839–848.
- CROCKETT, K.; BANDAR, Z.; MCLEAN, D. On the Optimization of T-norm parameters within Fuzzy Decision Trees. In: *2007 IEEE International Fuzzy Systems Conference*. IEEE, 2007. p. 1–6. ISBN 1-4244-1209-9. ISSN 1098-7584. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4295348>>.
- Da Ronco, C. C.; BENINI, E. A Simplex Crossover based evolutionary algorithm including the genetic diversity as objective. *Applied Soft Computing*, v. 13, n. 4, p. 2104–2123, abr. 2013. ISSN 15684946.
- De Jong, K. A. *Evolutionary computation: a unified approach*. [S.l.]: MIT press, 2006.
- DEB, K. et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 2, p. 182–197, abr. 2002.
- DEB, K. et al. Scalable multi-objective optimization test problems. In: *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*. [S.l.]: IEEE, 2002. v. 1, p. 825–830. ISBN 0-7803-7282-4. ISSN 10706631.
- DELGADO, M. R.; Von Zuben, F.; GOMIDE, F. Multi-objective decision making: towards improvement of accuracy, interpretability and design autonomy in hierarchical genetic fuzzy systems. In: *Fuzzy Systems, 2002. FUZZ-IEEE'02. Proceedings of the 2002 IEEE International Conference on*. [S.l.: s.n.], 2002. v. 2, p. 1222–1227.
- Di Nuovo, A. G.; CATANIA, V. Linguistic Modifiers to Improve the Accuracy-Interpretability Trade-Off in Multi-Objective Genetic Design of Fuzzy Rule Based Classifier Systems. In: *2009 Ninth International Conference on Intelligent Systems Design and Applications*. [S.l.]: IEEE, 2009. p. 128–133.
- DIAS, A.; VASCONCELOS, J. de. Multiobjective genetic algorithms applied to solve optimization problems. *IEEE Transactions on Magnetics*, v. 38, n. 2, p. 1133–1136, mar. 2002.
- DUBOIS, D.; PRADE, H. Soft computing, fuzzy logic, and artificial intelligence. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, Springer, v. 2, n. 1, p. 7–11, 1998.
- EIBEN, A. E. et al. Parameter control in evolutionary algorithms. In: *Parameter setting in evolutionary algorithms*. [S.l.]: Springer, 2007. p. 19–46.
- EVSUKOFF, A. G. et al. *Design of interpretable fuzzy rule-based classifiers using spectral analysis with structure and parameters optimization*. 2009. 857–881 p.

- FAZENDEIRO, P.; Valente de Oliveira, J.; PEDRYCZ, W. A multiobjective design of a patient and anaesthetist-friendly neuromuscular blockade controller. *IEEE Transactions on Bio-medical Engineering*, v. 54, n. 9, p. 1667–1678, set. 2007.
- FAZZOLARI, M. et al. A Review of the Application of Multiobjective Evolutionary Fuzzy Systems: Current Status and Further Directions. *Fuzzy Systems, IEEE Transactions on*, v. 21, n. 1, p. 45–65, 2013. ISSN 1063-6706.
- FONSECA, C.; FLEMING, P. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. *Proceedings of the Conference on Genetic Algorithms*, p. 416–423, 1993. ISSN 14639076. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.9077>>.
- GACTO, M.; ALCALÁ, R.; HERRERA, F. Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures. *Information Sciences*, v. 181, n. 20, p. 4340–4360, out. 2011.
- GACTO, M. J.; ALCALÁ, R.; HERRERA, F. Adaptation and application of multi-objective evolutionary algorithms for rule reduction and parameter tuning of fuzzy rule-based systems. *Soft Computing*, v. 13, n. 5, p. 419–436, ago. 2008.
- GACTO, M. J.; ALCALA, R.; HERRERA, F. Handling High-Dimensional Regression Problems by Means of an Efficient Multi-Objective Evolutionary Algorithm. In: *Intelligent Systems Design and Applications, 2009. ISDA '09. Ninth International Conference on*. [S.l.: s.n.], 2009. p. 109–114.
- GACTO, M. J.; ALCALÁ, R.; HERRERA, F. Integration of an Index to Preserve the Semantic Interpretability in the Multiobjective Evolutionary Rule Selection and Tuning of Linguistic Fuzzy Systems. *IEEE Transactions on Fuzzy Systems*, v. 18, n. 3, p. 515–531, jun. 2010.
- GACTO, M. J.; ALCALA, R.; HERRERA, F. Evolutionary Multi-Objective Algorithm to effectively improve the performance of the classic tuning of fuzzy logic controllers for a heating, ventilating and Air Conditioning system. In: *Genetic and Evolutionary Fuzzy Systems (GEFS), 2011 IEEE 5th International Workshop on*. [S.l.: s.n.], 2011. p. 73–80.
- GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. [S.l.]: Addison-Wesley, 1989. 412 p.
- GONZÁLEZ, A.; PÉREZ, R. SLAVE: a genetic learning system based on an iterative approach. *IEEE Transactions on Fuzzy Systems*, v. 7, n. 2, p. 176–191, abr. 1999.
- GREENE, D. P.; SMITH, S. F. Competition-based induction of decision models from examples. *Machine Learning*, v. 13, p. 229–257, 1993. ISSN 08856125.
- GUILLAUME, S. Designing fuzzy inference systems from data: An interpretability-oriented review. *IEEE Transactions on Fuzzy Systems*, v. 9, n. 3, p. 426–443, jun. 2001.
- GUILLAUME, S.; CHARNOMORDIC, B. Generating an Interpretable Family of Fuzzy Partitions From Data. *IEEE Transactions on Fuzzy Systems*, v. 12, n. 3, p. 324–335, jun. 2004.
- HÁJEK, J.; SZÖLLÖS, A.; VSÍSTEK, J. A new mechanism for maintaining diversity of Pareto archive in multi-objective optimization. *Advances in Engineering Software*, v. 41, n. 7-8, p. 1031–1057, jul. 2010. ISSN 09659978.

- HAJELA, P.; LIN, C. Y. Genetic search strategies in multicriterion optimal design. *Structural Optimization*, v. 4, n. 2, p. 99–107, jun. 1992.
- HAUPT, R. L.; HAUPT, S. E. *Practical genetic algorithms*. [S.l.]: John Wiley & Sons, 2004.
- HERRERA, F. Genetic fuzzy systems: taxonomy, current research trends and prospects. *Evolutionary Intelligence*, v. 1, n. 1, p. 27–46, jan. 2008.
- HOLLAND, J. H. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. [S.l.]: University of Michigan Press, 1975. 183 p.
- HOLLAND, J. H.; REITMAN, J. S. Cognitive systems based on adaptive algorithms. *Cognitive Systems*, p. 49, 1978. ISSN 01635719.
- HONG, T.-P. H. T.-P. et al. Learning Membership Functions in Takagi-Sugeno Fuzzy Systems by Genetic Algorithms. *2009 First Asian Conference on Intelligent Information and Database Systems*, 2009.
- HORN, J.; NAFPLIOTIS, N.; GOLDBERG, D. A niched Pareto genetic algorithm for multiobjective optimization. In: *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*. [S.l.]: IEEE, 1994. p. 82–87.
- ISHIBUCHI, H.; NAKASHIMA, Y.; NOJIMA, Y. Performance evaluation of evolutionary multiobjective optimization algorithms for multiobjective fuzzy genetics-based machine learning. *Soft Computing*, v. 15, p. 2415–2434, 2011. ISSN 14327643.
- ISHIBUCHI, H.; NOJIMA, Y. Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning. *International Journal of Approximate Reasoning*, v. 44, n. 1, p. 4–31, jan. 2007.
- ISHIBUCHI, H.; NOJIMA, Y. Difficulties in choosing a single final classifier from non-dominated solutions in multiobjective fuzzy genetics-based machine learning. In: *Proceedings of the 2013 Joint IFSA World Congress and NAFIPS Annual Meeting, IFSA/NAFIPS 2013*. [S.l.: s.n.], 2013. p. 1203–1208. ISBN 9781479903474.
- ISHIBUCHI, H.; YAMAMOTO, T. Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. *Fuzzy Sets and Systems*, v. 141, n. 1, p. 59–88, jan. 2004.
- ISHIBUCHI, H.; YAMAMOTO, T. Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. In: *Fuzzy Sets and Systems*. [S.l.: s.n.], 2004. v. 141, p. 59–88. ISSN 01650114.
- ISHIBUCHI, H. et al. Two-objective solution set optimization to maximize hypervolume and decision space diversity in multiobjective optimization. In: *IEEE. Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), 2012 Joint 6th International Conference on*. [S.l.], 2012. p. 1871–1876.
- JANG, J.-S. R.; SUN, C.-T.; MIZUTANI, E. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. [S.l.]: Prentice Hall, 1997. 614 p.

- JANG, J.-S. R.; SUN, C.-T.; MIZUTANI, E. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. [s.n.], 1997. 614 p. ISBN 978-0132610667. Disponível em: <http://www.amazon.com/Neuro-Fuzzy-Soft-Computing-Computational-Intelligence/dp/0132610663/ref=sr_1_1?s=books&ie=UTF8&qid=1344453390&sr=1-1&keywords=Neuro+Fuzzy+and+Soft+Computing>.
- JEE, T. L. et al. Building fuzzy inference systems with similarity reasoning: NSGAI-based fuzzy rule selection and evidential functions. In: *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*. [S.l.: s.n.], 2014. p. 2192–2197.
- KANDEL, A. *Fuzzy expert systems*. FL: USA: CRC Press, Inc., 1992.
- KAYA, M.; ALHAJJ, R. A clustering algorithm with genetically optimized membership functions for fuzzy association rules mining. *The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ '03.*, v. 2, 2003.
- KAYLANI, A. et al. An adaptive multiobjective approach to evolving ART architectures. *IEEE Transactions on Neural Networks*, v. 21, n. 4, p. 529–550, abr. 2010.
- KIM, D.; CHOI, Y.-S.; LEE, S.-Y. An accurate COG defuzzifier design using Lamarckian co-adaptation of learning and evolution. *Fuzzy Sets and Systems*, v. 130, n. 2, p. 207–225, set. 2002. ISSN 01650114. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0165011401001671>>.
- KISSI, M. et al. Determination of fuzzy logic membership functions using genetic algorithms: application to olfaction. In: *Signal Processing and Information Technology, 2003. ISSPIT 2003. Proceedings of the 3rd IEEE International Symposium on*. [S.l.: s.n.], 2003. p. 616–619.
- KITA, H. et al. Multi-objective Optimization by Means of the Thermodynamical Genetic Algorithm. *Lecture Notes in Computer Science*, p. 504–512, 1996.
- KLIR, G. J. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. [S.l.]: Pearson Education Taiwan Limited, 1995. 574 p.
- KNOWLES, J.; CORNE, D. The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation. In: *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99*. [S.l.]: IEEE, 1999. v. 1, p. 98–105.
- KOSKO, B. *Neural networks and fuzzy systems: a dynamical systems approach to machine intelligence*. [S.l.]: Prentice-Hall International, 1992. 449 p.
- KOZA, J. Genetic Programming: on the Programming of Computers by Means of Natural Selection. *Notes*, p. 1–187, 1992.
- KUSIAK, A. Evolutionary computation and data mining. In: GOPALAKRISHNAN, B.; GUNASEKARAN, A. (Ed.). *Proc. SPIE 4192, Intelligent Systems in Design and Manufacturing III, 1*. [S.l.: s.n.], 2000. v. 4192, p. 1–10.
- KUWAJIMA, I.; ISHIBUCHI, H.; NOJIMA, Y. Effectiveness of designing fuzzy rule-based classifiers from Pareto-optimal rules. In: *IEEE International Conference on Fuzzy Systems*. [S.l.: s.n.], 2008. p. 1185–1192. ISBN 9781424418190. ISSN 10987584.

- LAUMANNNS, M.; RUDOLPH, G.; SCHWEFEL, H.-P. A spatial predator-prey approach to multi-objective optimization: A preliminary study. In: EIBEN, A. E. et al. (Ed.). *Parallel Problem Solving from Nature-PPSN V*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, (Lecture Notes in Computer Science, v. 1498). p. 241–249.
- LAUMANNNS, M. et al. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary computation*, v. 10, n. 3, p. 263–82, jan. 2002. ISSN 1063-6560.
- LEE, C. Fuzzy logic in control systems: fuzzy logic controller. I. *IEEE Transactions on Systems, Man and Cybernetics*, v. 20, n. 2, p. 404–418, 1990.
- LEE, C. Fuzzy logic in control systems: fuzzy logic controller. II. *IEEE Transactions on Systems, Man and Cybernetics*, v. 20, n. 2, p. 419–435, 1990.
- LI, M.; YANG, S.; LIU, X. Diversity Comparison of Pareto Front Approximations in Many-Objective Optimization. *Cybernetics, IEEE Transactions on*, v. 44, n. 12, p. 2568–2584, 2014. ISSN 2168-2267.
- LICHMAN, M. *UCI Machine Learning Repository*. 2013. Disponível em: <<http://archive.ics.uci.edu/ml>>.
- LUKASIEWICZ, J. *philosophical remarks on many-valued systems of propositional logic*. [S.l.]: North Holland, 1970.
- MAMDANI, E.; ASSILIAN, S. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, v. 7, n. 1, p. 1–13, jan. 1975.
- MÁRQUEZ, A.; MÁRQUEZ, F. A.; PEREGRÍN, A. Cooperation between the Inference System and the Rule Base by Using Multiobjective Genetic Algorithms. In: *H AIS*. [S.l.: s.n.], 2008. p. 739–746.
- MARQUEZ, A. A.; MARQUEZ, F. A.; PEREGRIN, A. An efficient multi-objective evolutionary adaptive conjunction for high dimensional problems in linguistic fuzzy modelling. In: *Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on*. [S.l.: s.n.], 2012. p. 1–8. ISSN 1098-7584.
- MARQUEZ, F. A.; PEREGRIN, A.; HERRERA, F. Cooperative Evolutionary Learning of Linguistic Fuzzy Rules and Parametric Aggregation Connectors for Mamdani Fuzzy Systems. *IEEE Transactions on Fuzzy Systems*, v. 15, n. 6, p. 1162–1178, dez. 2007. ISSN 1063-6706.
- MARTIN, D. et al. QAR-CIP-NSGA-II: A new multi-objective evolutionary algorithm to mine quantitative association rules. *Information Sciences*, v. 258, n. 0, p. 1–28, 2014. ISSN 0020-0255. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0020025513006427>>.
- MENCAR, C.; CASTELLANO, G.; FANELLI, A. M. Some fundamental interpretability issues in fuzzy modeling. In: *Proceedings of the Joint 4th Conference of the European Society for Fuzzy Logic and Technology and the 11th Rencontres Francophones sur la Logique Floue et ses Applications*. [S.l.: s.n.], 2005. p. 100–105.
- MENCAR, C.; FANELLI, A. Interpretability constraints for fuzzy information granulation. *Information Sciences*, v. 178, n. 24, p. 4585–4618, dez. 2008.

- MICHALEWICZ, Z. *Genetic Algorithms + Data Structures = Evolution Programs*. [S.l.]: Springer Science & Business Media, 1996. 387 p.
- MILLER, G. A. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review*, v. 63, n. 2, p. 81–97, 1956.
- MITCHELL, M. *An introduction to genetic algorithms*. [S.l.]: MIT Press, 1998. 209 p.
- NGUYEN, C.; GANESH, C.; GONG, K. A fuzzy logic-based intelligent controller for contact management data integration. In: *Fuzzy Systems Proceedings. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*. [S.l.]: IEEE, 1998. v. 2, p. 879–884.
- NOJIMA, Y.; ISHIBUCHI, H. Interactive genetic fuzzy rule selection through evolutionary multiobjective optimization with user preference. In: *2009 IEEE Symposium on Computational Intelligence in Milti-Criteria Decision-Making*. [S.l.]: IEEE, 2009. p. 141–148.
- PEDRYCZ, W.; GOMIDE, F. *An Introduction to Fuzzy Sets: Analysis and Design*. [S.l.]: MIT Press, 1998. (A Bradford book).
- PIRES, M. G.; CARMARGO, H. d. A. Genetic learning and optimization of fuzzy sets in fuzzy rule-based system. In: *IEEE. Information Reuse and Integration, 2004. IRI 2004. Proceedings of the 2004 IEEE International Conference on*. [S.l.], 2004. p. 623–628.
- PULKKINEN, P. A Multiobjective Genetic Fuzzy System for Obtaining Compact and Accurate Fuzzy Classifiers with Transparent Fuzzy Partitions. In: *2009 International Conference on Machine Learning and Applications*. [S.l.]: IEEE, 2009. p. 89–94.
- PULKKINEN, P.; KOIVISTO, H. Fuzzy classifier identification using decision tree and multiobjective evolutionary algorithms. *International Journal of Approximate Reasoning*, v. 48, n. 2, p. 526–543, jun. 2008.
- REY, M. I. et al. Selection of rules by orthogonal transformations and genetic algorithms to improve the interpretability in fuzzy rule based systems. In: *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on*. [S.l.: s.n.], 2013. p. 1–8. ISSN 1098-7584.
- RUDOLPH, G. Evolutionary Search under Partially Ordered Fitness Sets. In: *Proceedings of the International Symposium on Information Science Innovations in Engineering of Natural and Artificial Intelligent Systems*. [S.l.: s.n.], 2001. p. 818–822.
- RUSPINI, E. H. A new approach to clustering. *Information and Control*, v. 15, n. 1, p. 22–32, jul. 1969.
- SAATY, T.; OZDEMIR, M. Why the magic number seven plus or minus two. *Mathematical and Computer Modelling*, v. 38, n. 3, p. 233–244, ago. 2003.
- SáEZ, D.; CORTÉS, C. E.; nEZ, A. N. Hybrid adaptive predictive control for the multi-vehicle dynamic pick-up and delivery problem based on genetic algorithms and fuzzy clustering. *Computers & Operations Research*, v. 35, n. 11, p. 3412–3438, nov. 2008.
- SÁNCHEZ, L.; COUSO, I.; CASILLAS, J. Genetic learning of fuzzy rules based on low quality data. *Fuzzy Sets and Systems*, v. 160, p. 2524–2552, 2009. ISSN 01650114.

- SCHAFFER, J. Multiple objective optimization with vector evaluated genetic algorithms. In: *Proceedings of the 1st international Conference on Genetic Algorithms*. [S.l.: s.n.], 1985. p. 93–100.
- SMITH, S. F. *No Title*. Tese (Doutorado) — University of Pittsburgh., 1980.
- SOLOW, A. R.; POLASKY, S. Measuring biological diversity. *Environmental and Ecological Statistics*, v. 1, n. 2, p. 95–103, jun. 1994. ISSN 1352-8505.
- SRINIVAS, N.; DEB, K. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, v. 2, n. 3, p. 221–248, set. 1994.
- STAVRAKOUDIS, D. G.; THEOCHARIS, J. B. A genetic fuzzy rule-based classifier for land cover image classification. In: *2009 IEEE International Conference on Fuzzy Systems*. [S.l.]: IEEE, 2009. p. 1677–1682.
- STAVRAKOUDIS, D. G.; THEOCHARIS, J. B. An evolutionary fuzzy classifier for satellite image classification. In: *2009 17th Mediterranean Conference on Control and Automation*. [S.l.]: IEEE, 2009. p. 383–388.
- SUGENO, M.; KANG, G. Structure identification of fuzzy model. *Fuzzy Sets and Systems*, v. 28, n. 1, p. 15–33, out. 1988.
- TAKAGI, T.; SUGENO, M. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions On Systems Man And Cybernetics*, v. 15, p. 116–132, 1985. ISSN 00189472. Disponível em: <<http://www.citeulike.org/user/cmowes/article/3928474>>.
- TAKAGI, T.; SUGENO, M. Fuzzy Identification of Systems and Its Applications to Modeling and Control. In: DUBOIS, D. J.; PRADE, H.; YAGER, R. R. (Ed.). *Readings in Fuzzy Sets for Intelligent Systems*. [S.l.]: Morgan Kaufmann Publishers Inc., 1993. v. 15, n. 1, p. 387–403.
- TARSKI, A.; MOSTOWSKI, A.; ROBINSON, R. M. *Undecidable theories*. [S.l.]: North-Holland Publications Inc., 1953. 98 p.
- TOFFOLO, A.; BENINI, E. Genetic diversity as an objective in multi-objective evolutionary algorithms. *Evolutionary computation*, v. 11, n. 2, p. 151–67, jan. 2003. ISSN 1063-6560.
- TSUKAMOTO, Y. An Approach to Fuzzy Reasoning Method. In: DUBOIS, D. J.; PRADE, H.; YAGER, R. R. (Ed.). *Readings in Fuzzy Sets for Intelligent Systems*. [S.l.]: Morgan Kaufmann Publishers Inc., 1993. p. 523–529.
- Van Veldhuizen, D. A. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. 272 p. Tese (Doctoral) — Air Force Institute of Technology, 1999.
- XIONG, N. Learning flexible structured linguistic fuzzy rules for mamdani fuzzy systems. In: *2009 IEEE International Conference on Fuzzy Systems*. [S.l.]: IEEE, 2009. p. 86–91. ISBN 978-1-4244-3596-8. ISSN 10987584.
- YANG, Y. et al. A hybrid evolutionary algorithm for finding pareto optimal set in multi-objective optimization. In: *Natural Computation (ICNC), 2011 Seventh International Conference on*. [S.l.: s.n.], 2011. v. 3, p. 1233–1236. ISSN 2157-9555.
- ZADEH, L. Fuzzy sets. *Information and Control*, v. 8, n. 3, p. 338–353, jun. 1965.

ZADEH, L. A. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3, n. 1, p. 28–44, 1973.

ZADOROJNIY, A. et al. Algorithms for finding maximum diversity of design variables in multi-objective optimization. *Procedia Computer Science*, Elsevier, v. 8, p. 171–176, 2012.

ZHANG, H.-x.; WANG, F.; ZHANG, B. Genetic optimization of fuzzy membership functions. In: *Wavelet Analysis and Pattern Recognition, 2009. ICWAPR 2009. International Conference on*. [S.l.: s.n.], 2009. p. 465–470.

ZHOU, S.-M.; GAN, J. Q. Low-level interpretability and high-level interpretability: a unified view of data-driven interpretable fuzzy system modelling. *Fuzzy Sets and Systems*, v. 159, n. 23, p. 3091–3131, dez. 2008.

ZITZLER, E.; LAUMANN, M.; THIELE, L. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*. [S.l.], 2001. 21 p.

ZITZLER, E.; THIELE, L. *An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach*. [S.l.], 1998. 43 p.

ZONG-YI, X. et al. Multi-objective fuzzy modeling using NSGA-II. In: *2008 IEEE International Conference on Cybernetics and Intelligent Systems, CIS 2008*. [S.l.: s.n.], 2008. ISBN 9781424416745.