UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

UMA ABORDAGEM PARA CONSTRUÇÃO DE SISTEMAS *FUZZY* BASEADOS EM REGRAS INTEGRANDO CONHECIMENTO DE ESPECIALISTAS E EXTRAÍDO DE DADOS

HELANO PÓVOAS DE LIMA

ORIENTADORA: PROFA. DRA. HELOISA DE ARRUDA CAMARGO

São Carlos – SP

Setembro/2015

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

UMA ABORDAGEM PARA CONSTRUÇÃO DE SISTEMAS *FUZZY* BASEADOS EM REGRAS INTEGRANDO CONHECIMENTO DE ESPECIALISTAS E EXTRAÍDO DE DADOS

HELANO PÓVOAS DE LIMA

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Inteligência Artificial

Orientadora: Profa. Dra. Heloisa de Arruda Camargo

São Carlos – SP

Setembro/2015

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária da UFSCar

L732ac

Lima, Helano Póvoas de.
Uma abordagem para construção de sistemas fuzzy baseados em regras integrando conhecimento de especialistas e extraído de dados / Helano Póvoas de Lima. -- São Carlos : UFSCar, 2015.
137 f.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2015.

1. Inteligência artificial. 2. Sistema Fuzzy. 3. Especialista de domínio. 4. Algoritmos genéticos. I. Título.

CDD: 006.3 (20ª)

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 Dissertação de Mestrado do candidato Helano Póvoas de Lima, realizada em 17/09/2015:

Profa. Dra. Heloisa de Arruda Camargo
UFSCar

Prof. Dr. Estevam Rafael Hruschka Junior
UFSCar

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

À minha esposa Andréia, minha filha Laura e minha mãe Uyara.

AGRADECIMENTOS

À Empresa Brasileira de Pesquisa Agropecuária - Embrapa, pela oportunidade e suporte financeiro.

Aos professores do PPGCC-UFSCar pelo conhecimento transmitido, em especial à professora Heloisa de Arruda Camargo pela oportunidade, confiança e direcionamento.

Aos pesquisadores Ériklis Nogueira e Urbano Gomes Pinto de Abreu, da Embrapa Pantanal, pela valiosa participação no estudo de caso desenvolvido durante este trabalho.

A verdadeira ciência ensina sobretudo a duvidar e a ser ignorante.

Miguel Unamuno

RESUMO

Historicamente, desde que Mamdani propôs seu modelo de sistema *fuzzy* baseado em regras, muita coisa mudou no processo de construção deste tipo de modelo. Durante muito tempo, os esforços de pesquisa foram direcionados à construção automática de sistemas precisos partindo de dados, tornando os sistemas *fuzzy* quase que meros aproximadores de função. Percebendo que esta abordagem fugia do conceito original da teoria *fuzzy*, mais recentemente, as atenções dos pesquisadores foram voltadas para a construção automática de modelos mais interpretáveis. Entretanto, tais modelos, embora interpretáveis, podem ainda não fazer sentido para o especialista. Este trabalho propõe uma abordagem interativa para construção de sistemas *fuzzy* baseados em regras, que visa ser capaz de integrar o conhecimento extraído de especialistas e induzido de dados, esperando contribuir para a solução do problema mencionado. A abordagem é composta por seis etapas. Seleção de atributos, definição das partições *fuzzy* das variáveis, definição da base de regras do especialista, aprendizado genético da base de regras, conciliação da base de regras e otimização genética da base de dados. As etapas de aprendizado e otimização utilizaram algoritmos genéticos multiobjetivo com operadores customizados para cada tarefa. Uma ferramenta de software foi implementada para subsidiar a aplicação da abordagem, oferecendo interfaces gráfica e de linha de comando, bem como uma biblioteca de software. A eficiência da abordagem foi avaliada por meio de um estudo de caso, onde um sistema *fuzzy* baseado em regras foi construído visando oferecer suporte à avaliação da aptidão reprodutiva de touros Nelore. O resultado foi comparado às metodologias de construção inteiramente manual e inteiramente automática, bem como a acurácia foi comparada a de algoritmos clássicos para classificação.

Palavras-chave: Sistema *fuzzy*, Metodologia para construção, Especialista de domínio, Algoritmo genético multiobjetivo

ABSTRACT

Historically, since Mamdani proposed his model of fuzzy rule-based system, a lot has changed in the construction process of this type of models. For a long time, the research efforts were directed towards the automatic construction of accurate models starting from data, making fuzzy systems almost mere function approximators. Realizing that this approach escaped from the original concept of fuzzy theory, more recently, researchers attention focused on the automatic construction of more interpretable models. However, such models, although interpretable, might not make sense to the expert. This work proposes an interactive methodology for constructing fuzzy rule-based systems, which aims to integrate the knowledge extracted from experts and induced from data, hoping to contribute to the solution of the mentioned problem. The approach consists of six steps. Feature selection, fuzzy partitions definition, expert rule base definition, genetic learning of rule base, rule bases conciliation and genetic optimization of fuzzy partitions. The optimization and learning steps used multiobjective genetic algorithms with custom operators for each task. A software tool was implemented to support the application of the approach, offering graphical and command line interfaces and a software library. The efficiency of the approach was evaluated by a case study where a fuzzy rule-based system was constructed in order to offer support to the evaluation of reproductive fitness of Nelore bulls. The result was compared to fully manual and fully automatic construction methodologies, the accuracy was also compared to classical algorithms for classification.

Keywords: fuzzy system, construction methodology, domain expert, multiobjective genetic algorithm

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS

AGMO – *Algoritmo Genético Multiobjetivo*

AG – *Algoritmo Genético*

BC – *Base de Conhecimento*

BD – *Base de Dados*

BR – *Base de Regras*

CART – *Árvore de Classificação e Regressão*

CF – *Conjunto Fuzzy*

FDT – *Árvore de Decisão Fuzzy*

MI – *Mecanismo de Inferência*

PFF – *Partição Fuzzy Forte*

RCD – *Regras do Conjunto de Dados*

REQM – *Raiz quadrada do Erro Quadrático Médio*

RE – *Regras do Especialista*

SFBR – *Sistema Fuzzy Baseado em Regras*

SFGMO – *Sistema Fuzzy Genético Multiobjetivo*

SFG – *Sistema Fuzzy Genético*

TCC – *Taxa de Classificação Correta*

TCF – *Teoria de Conjuntos Fuzzy*

UD – *Universo de Discurso*

SUMÁRIO

Capítulo 1

INTRODUÇÃO

As técnicas de modelagem de sistemas baseadas em inteligência computacional, como redes neurais artificiais, algoritmos genéticos, modelos gráficos probabilísticos, lógica *fuzzy* e mais recentemente, a combinação entre elas, assumem papel de destaque e são foco de intensa pesquisa, visto serem capazes de tratar problemas complexos, com muitas variáveis, que são difíceis de solucionar por métodos clássicos (EBERHART; SHI, 2007; HERRERA, 2008).

Desde que Lofti A. Zadeh fundamentou a teoria dos Conjuntos Fuzzy (CF) e a lógica *fuzzy*, bastante tem se estudado e aplicado com esse ferramental teórico. De fato, todo um novo ramo da computação, denominado Computação Flexível (*Soft Computing*), começou a se desenvolver com base nos formalismos propostos, procurando aprofundar as metodologias que visam explorar a tolerância à imprecisão e à incerteza, a fim de alcançar tratabilidade, robustez e soluções de baixo custo (ZADEH, 1994a, 1994b).

Sistemas de inferência baseados em lógica *fuzzy* têm sido aplicados a diversas áreas, como engenharia, modelagem, controle e várias outras, tornando-se uma das principais aplicações da teoria de conjuntos *fuzzy*. Seu sucesso deve-se à habilidade de tais sistemas em modelar o conhecimento baseando-se em termos linguísticos bem como à boa capacidade de generalização apresentada por eles (CORDÓN et al., 2004; GUILLAUME; CHARNOMORDIC, 2012; GUILLAUME; MAGDALENA, 2006).

O emprego da lógica *fuzzy* na modelagem e auxílio à tomada de decisão tomou impulso com o trabalho de Mamdani e Assilian (1975), o qual propôs um sistema *fuzzy* controlador para sintetizar o processo de tomada de decisão de um operador industrial habilitado, adotando um processo de decisão baseado em regras *fuzzy*. Nessas regras, tanto o antecedente quanto o consequente são compostos por proposições que definem valores de variáveis como termos linguísticos, expressos por meio de CF. Tais sistemas são chamados *Sistemas Fuzzy Baseados*

em Regras (SFBR) (KLIR; YUAN, 1995).

Uma característica notavelmente desejada dos SFBR é sua capacidade em "explicar" a elaboração do resultado a partir dos valores de entrada fornecidos. Esta interpretabilidade é fundamental para o entendimento dos processos sendo modelados e levam a modelos mais confiáveis e inteligíveis (ANTONELLI et al., 2010).

A modelagem por meio da extração de conhecimento diretamente do especialista [1] foi a primeira abordagem utilizada para construção de SFBR inspirada pelo trabalho de Mamdani. Porém, tal tarefa não é trivial, dependendo bastante da habilidade de um engenheiro de conhecimento e do especialista do domínio em formalizar o conhecimento empírico. Outro problema é que, na maioria das vezes, o especialista consegue representar apenas as tendências do sistema e alguns casos particulares, tendo um conhecimento parcial ou incompleto. Estes fatores podem levar a sistemas com considerável perda de acurácia e, consequentemente, baixa usabilidade.

Na metade da década de 80, as primeiras abordagens para construção de SFBR utilizando conhecimento extraído automaticamente de dados começaram a surgir (TAKAGI; SUGENO, 1985), tornando a técnica muito popular e iniciando uma nova era na modelagem e aplicação destes sistemas, principalmente na área de controle, visto a capacidade dos SFBR como aproximadores. Porém, estes modelos focavam principalmente na acurácia, tornado-os cada vez mais parecidos com modelos "caixa preta" devido às suas características majoritariamente numéricas, distanciando-os dos especialistas que poderiam não ser capazes de ver significado em seus parâmetros. Mesmo quando o conhecimento de especialistas de domínio de conhecimento era utilizado neste tipo de modelagem, ele se restringia a um conhecimento inicial, de onde um processo de aprendizado de máquina por meio de dados continuaria a construção do SFBR, sem preocupações com a interpretabilidade do resultado (GUILLAUME; MAGDALENA, 2006).

Apenas o uso do formalismo *fuzzy* em si não garante que um sistema seja interpretável (GUILLAUME, 2001). No final dos anos 90 a comunidade científica identificou que a abordagem corrente havia se distanciado dos conceitos originais propostos por Zadeh, que buscavam preencher a lacuna existente entre o entendimento humano e o processamento da máquina. As atenções então voltaram-se novamente aos atributos de interpretabilidade dos sistemas *fuzzy*, direcionando um grande número de trabalhos ao propósito do aprendizado automatizado, por meio de dados, de sistemas *fuzzy* interpretáveis.

Embora a maioria destes trabalhos seja motivada e enfatize a questão da participação do especialista ser importante durante o processo de construção de um SFBR, visando garantir que o

[1] Entende-se por especialista, alguém com profundo conhecimento adquirido por experiência ao longo do tempo, sobre os mecanismos de funcionamento de um determinado fenômeno ou processo.

produto do aprendizado automático seja próximo de sua linguagem, tais iniciativas preocupam-se, principalmente, com a legibilidade do modelo. Entretanto, mesmo que um ser humano consiga ler e entender os termos linguísticos e as regras contidas neste modelos, nada garante que o conhecimento expresso nele seja compatível com o que um especialista de domínio esperaria encontrar.

São poucos os trabalhos encontrados na literatura que propõem uma abordagem que permita integrar o conhecimento do especialista e o conhecimento extraído de dados em um mesmo processo, até o produto final. Abordagens que buscam esta integração, podem ser vistas em (ALONSO; MAGDALENA; GUILLAUME, 2008; ALONSO; MAGDALENA, 2010, 2011; PANCHO et al., 2013; PANCHO; ALONSO; MAGDALENA, 2013; COULON-LEROY et al., 2013; GUILLAUME, 2001; GUILLAUME; MAGDALENA, 2006; GUILLAUME; CHARNOMORDIC, 2012).

Os benefícios da participação do especialista no processo de construção são muitos. Desde que o processo seja guiado por algumas restrições, o modelo resultante, possivelmente, terá uma semântica correta e próxima do conhecimento do especialista, permitindo que ele não só seja capaz de lê-lo, como também de fazer julgamentos sobre o comportamento das saídas inferidas, facilitando os processos de validação, manutenção e aplicação do modelo resultante.

Métodos para a integração das preferências do especialista no processo de construção de SFBRs têm sido considerados um problema em aberto de grande interesse (CORDÓN, 2011; GUILLAUME; CHARNOMORDIC, 2012). Em um âmbito mais geral ainda, o especialista pode ser considerado um elo para o sucesso e impacto da área de pesquisa em aprendizado de máquina, visto ser ele um dos principais responsáveis pela aplicação do ferramental desenvolvido em problemas do mundo real, fornecendo o *feedback* de onde se encontra a demanda por melhorias (WAGSTAFF, 2012).

Interpretabilidade é sempre desejável em um modelo, especialmente em sistemas baseados em conhecimento onde a interação humana é necessária, particularmente, em sistemas de suporte à decisão em áreas críticas como medicina, engenharia, finanças, etc. Em tais sistemas, esta participação pode ser considerada mandatória, pois os usuários precisam entender o modelo e sentir confiança em suas sugestões ou, então, não as aceitarão e o sistema não alcançará seu propósito (ALONSO; MAGDALENA; GUILLAUME, 2008).

Ao longo dos anos, algumas das técnicas para aprendizado e otimização de SFBR a partir de dados mais bem sucedidas têm sido as baseadas em algoritmos genéticos (AG), visto a habilidade de tais algoritmos em explorar vastos espaços de busca por soluções adequadas, requerendo apenas poder computacional para tal (CORDÓN et al., 2004; HERRERA, 2008). Além

disso, AG ainda hoje permanecem como uma das poucas opções disponíveis quanto a levar em conta decisões de projeto, permitindo que especialistas decidam quais componentes querem fixar e quais querem evoluir de acordo com medidas de performance (HERRERA, 2008). Os Algoritmos Genéticos Multiobjetivo (AGMO), uma das principais tendências de pesquisas na área, têm-se mostrado adequados para tratar o problema de geração automática de SFBR por possuírem um processo evolutivo que busca o balanceamento entre objetivos contraditórios (CORDÓN, 2011; FAZZOLARI et al., 2013). Na geração dos SFBR essa questão é pertinente, uma vez que duas das características mais relevantes e desejáveis do sistema gerado, a acurácia e a interpretabilidade, são objetivos contraditórios e devem, portanto, ser balanceados.

1.1 Objetivo

O objetivo deste trabalho é desenvolver uma abordagem interativa para construção de SFBRs, que seja capaz de integrar o conhecimento extraído de especialistas de domínio de conhecimento e aquele induzido de dados, aproveitando as vantagens que cada um pode oferecer durante as etapas deste processo. A abordagem manterá o especialista no controle das decisões nos níveis linguístico e semântico, enquanto um processo de aprendizado e otimização dos componentes do SFBR ficará a cargo de um AGMO e outras técnicas de aprendizado de máquina, utilizando dados.

1.2 Organização do trabalho

O restante deste trabalho está dividido da seguinte maneira: No Capítulo 2 serão abordados conceitos básicos relativos a sistemas *fuzzy*, como a teoria de conjuntos *fuzzy* e sistemas *fuzzy* baseados em regras. No Capítulo 3 serão abordados conceitos básicos relativos a computação evolutiva, como Algoritmos genéticos, sistemas fuzzy genéticos e sistemas fuzzy genéticos multiobjetivo. No Capítulo 4 serão discutidos aspectos da interpretabilidade na construção de SFBR. No Capítulo 5 será apresentada a abordagem proposta. No Capítulo 6 será apresentada a ferramenta desenvolvida para viabilizar a abordagem. No Capítulo 7 será apresentado um estudo de caso aplicando a abordagem proposta e, por fim, no Capítulo 8 serão apresentadas as considerações finais e trabalhos futuros.

Capítulo 2

SISTEMAS *Fuzzy*

2.1 Teoria de Conjuntos *Fuzzy*

Todas as definições apresentadas nesta seção podem ser encontradas de forma completa em (KLIR; YUAN, 1995; PEDRYCZ; GOMIDE, 1998; NICOLETTI; CAMARGO, 2004).

Segundo a teoria clássica de conjuntos, um elemento pertence ou não a um dado conjunto (conjuntos *crisp*). Não há valor intermediário de pertinência. Porém, no mundo real, tal conceito nem sempre se aplica, pois as fronteiras entre o que define um conjunto ou outro podem não ser precisamente definidas. Por exemplo, uma xícara de café com uma temperatura medindo 55°C pertence ao conjunto *Quente* ou *Morno*?

Zadeh (1965) fundamentou o conceito de CF, como sendo uma extensão da teoria clássica de conjuntos, na qual cada elemento possui um grau de pertinência ao conjunto, que usualmente é um valor entre 'zero' e 'um'. Este conceito subsidiou um conjunto de operações análogo ao presente na teoria de conjuntos clássica, dando origem à *Teoria de Conjuntos Fuzzy* (TCF). De forma semelhante à extensão do conceito de conjunto, Zadeh propôs também uma extensão da lógica multivalorada, definindo valores verdade linguísticos e criando assim a lógica *fuzzy*. Segundo o autor, tais conceitos permitem que sejam tratados problemas do mundo real, onde os critérios de pertinência e as fronteiras entre classes não são precisamente definidos (nebulosos ou difusos).

2.1.1 Conceitos Básicos da TCF

- **Função de pertinência:** Assim como a função característica de um conjunto *crisp* define a pertinência (1 se pertence ou 0 se não pertence) de um elemento do conjunto universo a um determinado conjunto, a função de pertinência de um conjunto *fuzzy* pode

ser abordada como uma função característica generalizada, que mapeia cada elemento do conjunto universo em um determinado valor, pertinente a um intervalo (usualmente o intervalo $[0,1]$), que reflete o *grau de pertinência* (grau de compatibilidade) do elemento ao conjunto *fuzzy* sendo definido.

- **Conjunto *Fuzzy*:** Um conjunto *fuzzy* A é definido em termos de um conjunto universo X, por meio de sua função de pertinência, que atribui a cada elemento $x \in X$, um número, $A(x) \in [0,1]$, que representa o grau de pertinência de x a A, como em (2.1).

$$A : X \to [0,1] \tag{2.1}$$

Outra notação comumente utilizada para funções de pertinência é mostrada em (2.2). Se X é um dado conjunto universo e A é um conjunto *fuzzy*, então sua função de pertinência (μ_A) pode ser notada como:

$$\mu_A : X \to [0,1] \tag{2.2}$$

- **Representação de funções de pertinência:** Quando conjuntos *fuzzy* são definidos em um universo finito, a representação analítica (por meio de uma função parametrizável) é recomendada. As famílias de funções parametrizáveis mais comumente utilizadas para representar conjuntos *fuzzy* são:

  - **Funções Triangulares:** definida pelos parâmetros a, m e b, sendo $a \leq m \leq b$, como descrito em (2.3) e representado na Figura 2.1:

$$A(x) = \begin{cases} 0 & se \quad x \leq a \\ \frac{x-a}{m-a} & se \quad x \in (a,m) \\ 1 & se \quad x = m \\ \frac{b-x}{b-m} & se \quad x \in (m,b) \\ 0 & se \quad x \geq b \end{cases} \tag{2.3}$$

  - **Funções Trapezoidais:** definida pelos parâmetros a, m, n e b, sendo $a \leq m < n \leq b$, como descrito em (2.4) e representado na Figura 2.2:

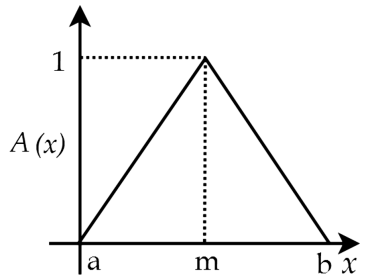


Figura 2.1: Representação de uma função de pertinência triangular.

$$A(x) = \begin{cases} 0 & se \quad x \leq a \\ \frac{x-a}{m-a} & se \quad x \in (a,m) \\ 1 & se \quad x \in [m,n] \\ \frac{b-x}{b-n} & se \quad x \in (n,b) \\ 0 & se \quad x \geq b \end{cases} \tag{2.4}$$

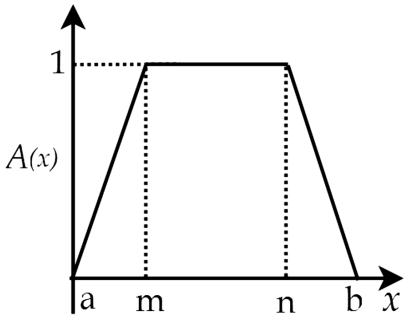


Figura 2.2: Representação de uma função de pertinência trapezoidal.

– **Funções Gaussianas:** definida pelos parâmetros m e k, sendo $k > 0$, como descrito em (2.5) e representado na Figura 2.3:

$$A(x) = e^{-k(x-m)^2} \tag{2.5}$$

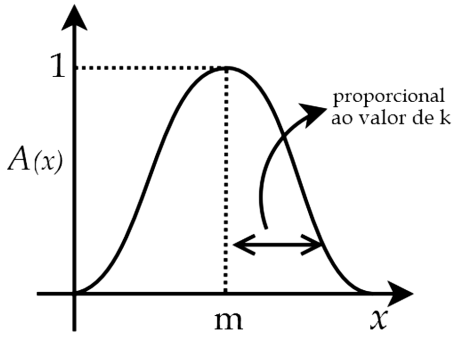


Figura 2.3: Representação de uma função de pertinência gaussiana.

– **Funções do tipo *S*:** definida pelos parâmetros a, m e b, como descrito em (2.6) e representado na Figura 2.4. O ponto $m = \frac{(a+b)}{2}$ é conhecido como cruzamento da função S.

$$A(x) = \begin{cases} 0 & se \quad x \leq a \\ 2\left(\frac{x-a}{b-a}\right)^2 & se \quad x \in (a,m] \\ 1-2\left(\frac{x-b}{b-a}\right)^2 & se \quad x \in (m,b] \\ 1 & se \quad x > b \end{cases} \tag{2.6}$$

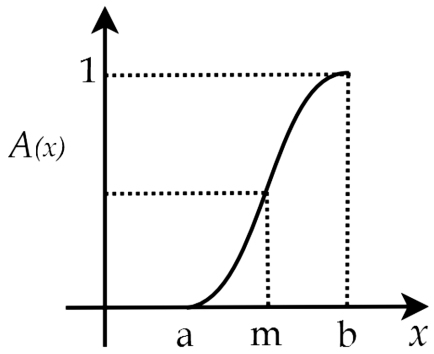


Figura 2.4: Representação de uma função de pertinência do tipo S.

2.1.2 Definições associadas a Conjuntos *Fuzzy*

Seja F um conjunto *fuzzy*, definido em um conjunto universo X:

- **Suporte:** O suporte de um conjunto *fuzzy* $F(S_F)$ é o conjunto numérico contendo todos os elementos de X com grau de pertinência diferente de 0 em F, como em (2.7) e mostrado na Figura 2.5.

$$S_F = \{x \in X \mid F(x) > 0\} \tag{2.7}$$

- **Núcleo:** O núcleo de um conjunto *fuzzy* $F(N_F)$ é o conjunto numérico contendo todos os elementos de X com grau de pertinência igual a 1 em F, como em (2.8) e mostrado na Figura 2.5.

$$N_F = \{x \in X \mid F(x) = 1\} \tag{2.8}$$

- **Altura:** A altura de um conjunto *fuzzy* $F(H_F)$ é definida como o maior grau de pertinência obtido por qualquer elemento do conjunto F, como em (2.9).

$$H_F = sup_x\{F(x) \mid x \in F\} \tag{2.9}$$

- **Normalidade:** Um conjunto *fuzzy* é considerado como conjunto normal se sua altura for igual a 1, como mostrado na Figura 2.5. Caso contrário ele é chamado de subnormal. Os conjuntos subnormais têm como núcleo um conjunto vazio.

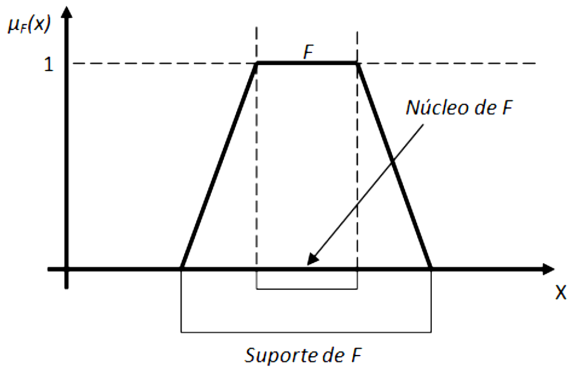


Figura 2.5: Representação do suporte e núcleo do conjunto *fuzzy* normal F.

2.1.3 Operações Padrão sobre Conjuntos *Fuzzy*

Como já mencionado, a TCF é uma extensão da teoria de conjuntos clássica, logo os operadores de complemento, união e interseção se estendem também. Nos conjuntos *fuzzy*, porém, estes operadores deixam de ser únicos e podem ser implementados por meio de classes de operadores. Dentre cada uma destas classes, existem os operadores padrão, que são os mais frequentemente utilizados.

Sejam A e B dois conjuntos *fuzzy* definidos no conjunto universo X, e $A(x)$ e $B(x)$ suas funções de pertinência, respectivamente.

- **União *fuzzy* padrão:** É definida pelo operador de máximo, como descrito em (2.10). A Figura 2.6 (a) representa os conjuntos A e B, enquanto a área não pontilhada na Figura 2.6 (b) representa $(A \cup B)$:

$$(A \cup B)(x) = \max[A(x), B(x)] \tag{2.10}$$

- **Intersecção *fuzzy* padrão:** É definida pelo operador de mínimo, como descrito em (2.11). A Figura 2.7 (a) representa os conjuntos A e B, enquanto a área não pontilhada na Figura 2.7 (b) representa $(A \cap B)$:

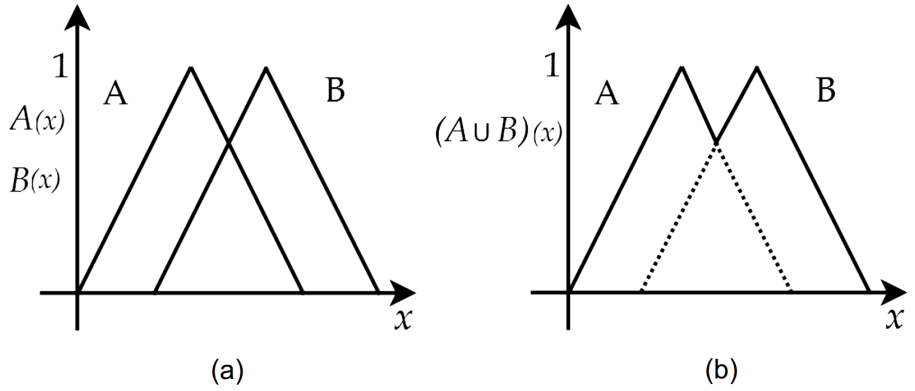


Figura 2.6: Representação da operação de união padrão entre os conjuntos A e B.

$$(A \cap B)(x) = \min[A(x), B(x)] \tag{2.11}$$

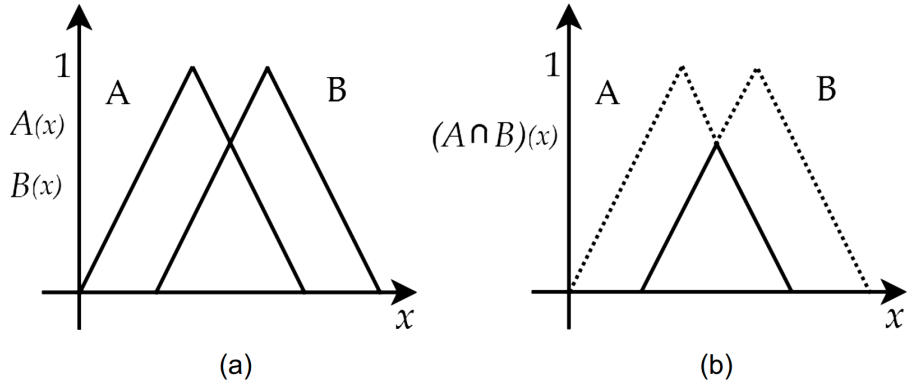


Figura 2.7: Representação da operação de intersecção padrão entre os conjuntos A e B.

- **Complemento *fuzzy* padrão:** É definida como descrito em (2.12). A Figura 2.8 (a) representa o conjuntos A, enquanto a área não pontilhada na Figura 2.8 (b) representa $\overline{A}$:

$$\overline{A}(x) = 1 - A(x) \tag{2.12}$$

2.1.4 Operações Generalizadas sobre Conjuntos *Fuzzy*

Para cada classe de operadores (união, intersecção e complemento) existe uma vasta quantidade de funções que se classificam como generalizações *fuzzy* dos operadores clássicos, assumindo formas diferentes dos operadores *fuzzy* padrão. Funções que se qualificam como intersecção são chamadas *t-normas*, enquanto as que se qualificam como união são chamadas *t-conormas* ou *s-normas*. Tais funções devem garantir que certas propriedades de operações entre conjuntos sejam satisfeitas, conforme definido a seguir:

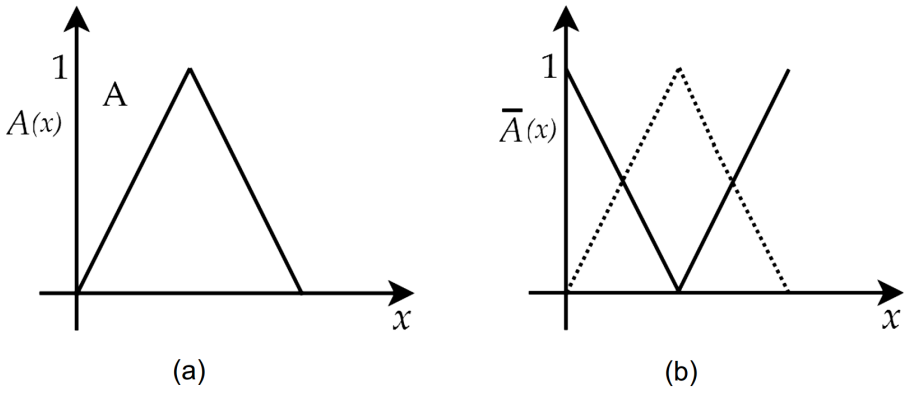


Figura 2.8: Representação da operação de complemento padrão do conjunto A.

- **T-normas:** São operações binárias definidas por $i : [0,1]^2 \to [0,1]$ que satisfazem os seguintes axiomas para todo a, b e $c \in [0,1]$:
  - ***Condição limite:*** $i(a,1) = a$.
  - ***Monotonicidade:*** $b \leq c$ implica $i(a,b) \leq i(a,c)$.
  - ***Comutatividade:*** $i(a,b) = i(b,a)$.
  - ***Associatividade:*** $i(a,i(b,c)) = i(i(a,b),c)$.

  As t-normas definidas a seguir, são as mais frequentemente utilizadas como intersecção *fuzzy*. Para todo $a,b \in [0,1]$:
  - ***Intersecção padrão:*** $i(a,b) = \min(a,b)$.
  - ***Produto algébrico:*** $i(a,b) = ab$.
  - ***Diferença limitada:*** $i(a,b) = \max(0, a+b-1)$.
  - ***Intersecção drástica:*** $i_{\min}(a,b) = \begin{cases} a & \text{quando } b = 1 \\ b & \text{quando } a = 1 \\ 0 & \text{caso contrário} \end{cases}$

- **T-conormas:** São operações binárias definidas por $u : [0,1]^2 \to [0,1]$ que satisfazem os seguintes axiomas para todo a, b e $c \in [0,1]$:
  - ***Condição limite:*** $u(a,0) = a$.
  - ***Monotonicidade:*** $b \leq c$ implica $u(a,b) \leq u(a,c)$.
  - ***Comutatividade:*** $u(a,b) = u(b,a)$.
  - ***Associatividade:*** $u(a,u(b,c)) = u(u(a,b),c)$.

As t-conormas definidas a seguir, são as mais frequentemente utilizadas como união *fuzzy*. Para todo $a, b \in [0,1]$:

- ***União padrão:*** $u(a,b) = \max(a,b)$.
- ***Soma algébrica:*** $u(a,b) = a + b - ab$.
- ***Soma limitada:*** $u(a,b) = \min(1, a+b)$.
- ***União drástica:*** $u_{\max}(a,b) = \begin{cases} a & \text{quando } b = 0 \\ b & \text{quando } a = 0 \\ 1 & \text{caso contrário} \end{cases}$

2.2 Sistemas *Fuzzy* Baseados em Regras

De maneira simplificada, pode-se considerar como sistema *fuzzy* qualquer sistema que faz uso da TCF para representar suas variáveis ou suas interações. As variáveis em um sistema *fuzzy* são chamadas *variáveis linguísticas*, pois seus valores são sentenças na forma de linguagem natural, por exemplo, temperatura, altura, velocidade, distância, etc. Tais variáveis são definidas por *termos linguísticos*, que são rótulos ou valores de uma variável linguística aos quais estão associados conjuntos *fuzzy*, por exemplo, frio, alto, rápido, longe, etc.

O universo de discurso (UD) de uma variável em um sistema *fuzzy* é o conjunto de todos os valores *crisp* que uma variável pode assumir (conjunto universo) e sobre o qual os conjuntos *fuzzy* são definidos. O processo de particionar o UD de uma variável em termos linguísticos, define uma *partição fuzzy*. A Figura 2.9 mostra um exemplo de partição *fuzzy*.

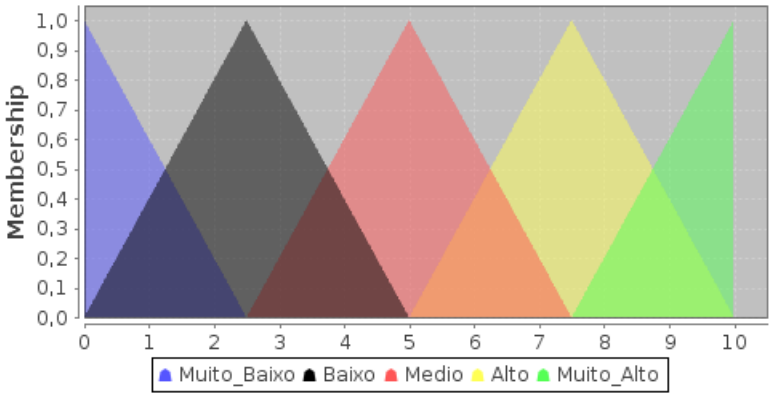


Figura 2.9: Exemplo de partição *fuzzy*.

Para esta proposta, o tipo específico de sistema *fuzzy* de interesse é o SFBR. As partes essenciais de um SFBR são a base de conhecimento (BC) e o mecanismo de inferência (MI).

A BC por sua vez é dividida entre a base de dados (BD) e a base de regras (BR). A estrutura genérica de um SFBR pode ser vista na Figura 2.10.

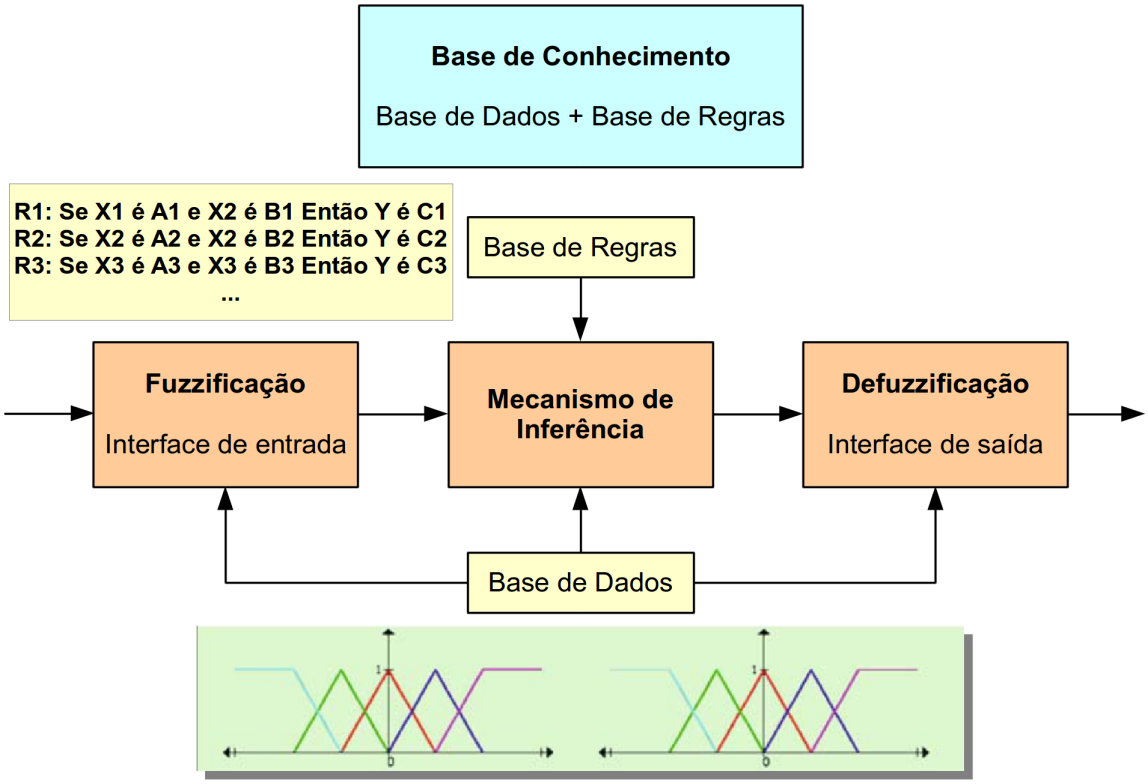


Figura 2.10: Estrutura de um SFBR (adaptada de (HERRERA, 2008)).

Na BD são descritos os termos linguísticos associados às variáveis do problema e suas respectivas funções de pertinência que descrevem sua semântica. Cada variável do problema tem associada uma partição *fuzzy* do seu universo, formada pelos CF associados com cada termo linguístico. Esta abordagem pode ser considerada um tipo de discretização para domínios contínuos onde é estabelecido uma função de pertinência para cada termo linguístico e há uma sobreposição entre eles.

A BR descreve um conjunto de regras *fuzzy* associadas ao problema em função das variáveis e dos termos linguísticos da BD. Ela desempenha um papel chave nos SFBR, pois é por meio das regras que o conhecimento é representado no sistema. O MI é capaz então, de processar as regras a partir de fatos conhecidos, de acordo com um dado método de raciocínio, fornecendo uma conclusão. O formato das regras *fuzzy* em um SFBR segue o padrão:

"SE *um conjunto de condições são satisfeitas* ENTÃO *um conjunto de consequências podem ser inferidas*"

O funcionamento do restante dos componentes de um SFBR depende do modelo de inferência adotado e será abordados mais adiante. Existem dois modelos principais de SFBR que

recebem mais destaque na literatura devido seu sucesso em aplicações práticas. Em ambos o antecedente das regras é formado por variáveis linguísticas e seus respectivos termos linguísticos definidos na BD, residindo no consequente das regras sua principal diferença, a saber:

- **Linguístico ou de Mamdani:** Os consequentes das regras também são formados por termos linguísticos da BD. A saída é formada a partir da união dos CF inferidos de cada regra ativada na BR, gerando um CF agregado, ao qual então é aplicado um processo para transformá-lo em um valor numérico de saída (MAMDANI; ASSILIAN, 1975; MAMDANI, 1977);
- **Funcional ou Takagi-Sugeno:** Os consequentes das regras são uma função polinomial aplicada aos valores de entrada. A saída é formada a partir da média ponderada do resultado da função de cada regra da BR. Os coeficientes de ponderação correspondem ao grau de ativação de cada regra. Esta abordagem aproxima um sistema não linear a partir de vários sistemas lineares (TAKAGI; SUGENO, 1985).

O modelo de inferência de interesse nesta proposta é o de Mamdani, visto ser mais intuitivo e adequado à intervenção humana, encaixando-se melhor ao contexto do problema sendo abordado. Seus passos básicos (YING, 2000) são descritos a seguir:

- **Fuzzificação:** É o processo matemático pelo qual o valor numérico (*crisp*) de uma variável de entrada é convertido para um valor de pertinência a um CF (termo linguístico) por meio da função de pertinência do mesmo. Este processo é melhor ilustrado na Figura 2.11 (b). No exemplo, observa-se que dado o valor de entrada $6,5$ para a variável X, seu grau de pertinência (*membership*) é de $0,25$ ao CF RUIM e de $0,75$ ao CF REGULAR;
- **Inferência:** É o processo de raciocínio por meio do qual são ativadas as regras da BR em que o antecedente possui CF (termos linguísticos) que tiveram grau de pertinência maior que zero na fuzzificação. Para cada regra ativada, os CF no consequente são então implicados de acordo com alguma operação aplicada aos graus de pertinência de cada CF no antecedente, que no método de inferência de Mamdani é o operador de mínimo (MIN). Tomando o exemplo da Figura 2.11, as regras R1 e R2 seriam ativadas, enquanto a R3 não seria (Figura 2.11 (a)).
- **Defuzzificação:** É o processo por meio do qual os CF das variáveis de saída das regras ativadas no processo de inferência da BR são agregados e, então, submetidos à um método para converter este resultado em um valor numérico (*crisp*). Para isso, várias expressões

matemáticas podem ser usadas, uma das mais adotadas é *centro de gravidade*, que consiste em dividir a área sob a função de pertinência do conjunto resultante da agregação em duas subáreas iguais. Tal processo é ilustrado na Figura 2.11 (c), onde a área em azul escuro corresponde à agregação dos CF da variável de saída Y das regras ativadas.

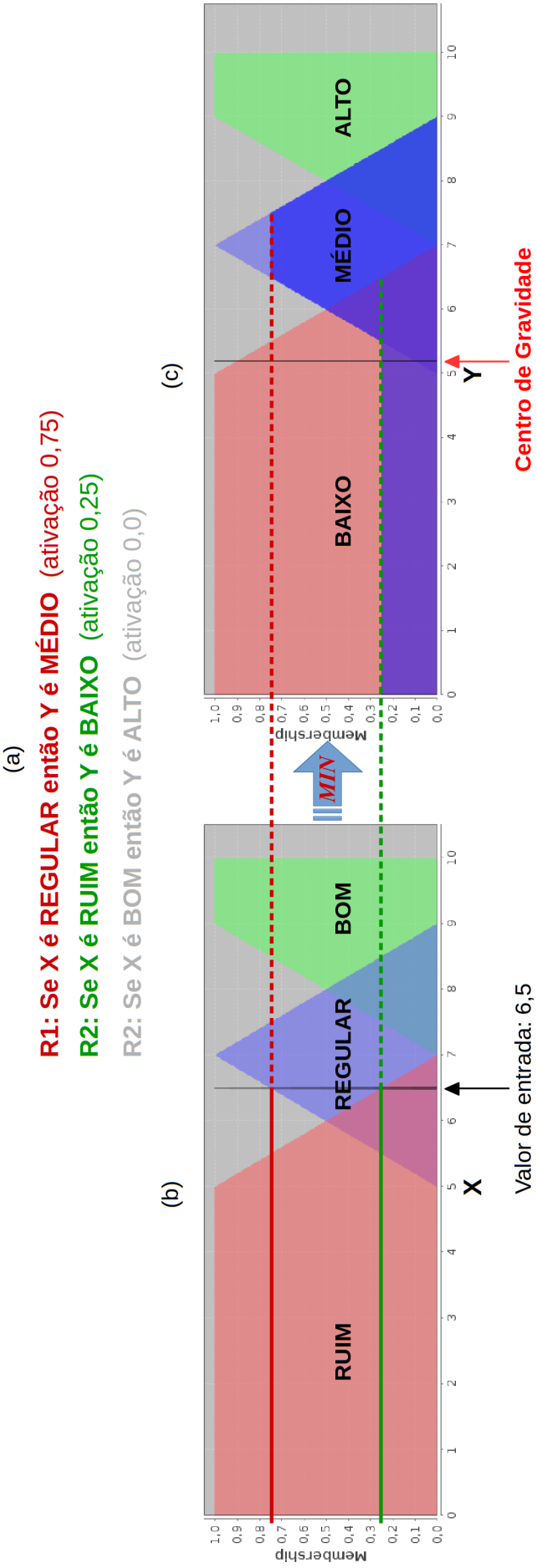


Figura 2.11: Exemplo do processo de inferência *fuzzy*.

Capítulo 3

COMPUTAÇÃO EVOLUTIVA

3.1 Algoritmos Genéticos

A Computação Evolucionária ou Evolutiva é um ramo da Computação cujo paradigma é inspirado nas teorias de seleção natural e evolução das espécies, propostas por Charles Darwin em seu livro *A Origem das Espécies* de 1859.

Uma das principais técnicas nesta linha de pesquisa são os AG, notoriamente reconhecidos por suas capacidades de busca e otimização. Alguns aspectos dos AG garantem que eles sejam bastante populares e vastamente aplicados em diversas áreas, como industrial, química, engenharia, biologia, etc. Dentre os aspectos, destacam-se a facilidade de codificação do problema, a não obrigatoriedade de conhecimento de parâmetros e informações auxiliares a respeito do problema e a capacidade de evitar os ótimos locais (EBERHART; SHI, 2007).

3.1.1 Conceitos Básicos de AG

- **Cromossomo ou Indivíduo:** Codifica uma solução possível para o problema através de uma cadeia finita de símbolos, que varia de tamanho conforme a quantidade de parâmetros do problema. Cada cromossomo tem atribuído um valor de aptidão que mede o quanto a solução que ele codifica é compatível com o resultado esperado;
- **Gene:** São os parâmetros do problema codificados. A junção de todos eles forma o cromossomo;
- **População:** É o conjunto de indivíduos candidatos à solução do problema. Possui tamanho fixo.

- **Geração:** Corresponde a um ciclo do AG, onde a população é transformada em uma outra, na busca de indivíduos mais aptos como solução;
- **Avaliação da Aptidão:** É uma função cujas variáveis são os genes de um cromossomo. Esta função deve ser capaz de avaliar o quão próximo um indivíduo (solução) está do resultado esperado;
- **Seleção:** É o processo de escolha dos indivíduos que darão origem a uma nova população (Pais). Os indivíduos com maior aptidão têm mais chances de serem selecionados, entretanto, o método escolhido deve dar a oportunidade de um indivíduo menos apto ser selecionado;
- **Recombinação ou Cruzamento:** É o processo de reprodução dos indivíduos selecionados (Pais), onde são combinadas suas características, a fim de formar novos indivíduos para a população (Filhos);
- **Mutação:** Consiste na alteração aleatória de algumas características dos indivíduos originados no cruzamento, a fim de manter a diversidade da população, buscando evitar a convergência para ótimos locais;
- **Atualização da População:** Consiste na junção dos Pais e dos Filhos para a formação da geração seguinte;
- **Critério de Parada:** É a condição que, quando satisfeita, faz o algoritmo finalizar a busca. Geralmente é uma quantidade definida de gerações e/ou um percentual de erro mínimo aceitável para a solução;
- **Finalização:** Consiste na solução alcançada, geralmente dada pelo indivíduo mais apto da população.

Alguns parâmetros comuns à maioria dos AG também merecem ser analisados, pois influenciam diretamente no seu comportamento e devem ser escolhidos de maneira apropriada ao problema estudado:

- **Tamanho da População:** Deve ser adequado ao espaço de busca do problema. Não deve ser nem pequeno, para garantir a diversidade genética dos indivíduos e consequentemente a capacidade do algoritmo de convergir para soluções globais ao invés de locais, e nem grande demais ao ponto de ter um custo computacional proibitivo;

- **Taxa de Cruzamento:** Quanto mais alta, maior a velocidade com que novas soluções são geradas. Não pode ser alta demais, para evitar que indivíduos com alta aptidão sejam perdidos, e nem baixa demais a ponto de prejudicar a convergência para a solução;
- **Intervalo de Geração:** Corresponde ao tamanho da parcela da população que será substituída a cada geração. Incorre nos mesmos problemas da taxa de cruzamento;
- **Taxa de Mutação:** Quanto mais alta, maior a capacidade de cobrir com eficiência o espaço de busca do problema; entretanto, uma taxa demasiadamente alta torna a busca aleatória.

3.1.2 Funcionamento de um AG

Pode-se definir os AG como algoritmos iterativos que implementam uma busca probabilística, paralela e estruturada sobre o espaço de busca de um problema. Por meio de uma população de indivíduos de número fixo, cada um representando uma possível solução, a cada geração do algoritmo, os indivíduos da população têm seus genes modificados por meio de cruzamento e/ou mutação, sendo então, selecionados conforme sua aptidão em solucionar o problema, para compor a população da geração seguinte. O ciclo se repete até que um critério de parada seja alcançado. O funcionamento geral de um AG é mostrado no fluxograma da Figura 3.1, onde t representa a geração atual e $P(t)$ representa a população da geração t.

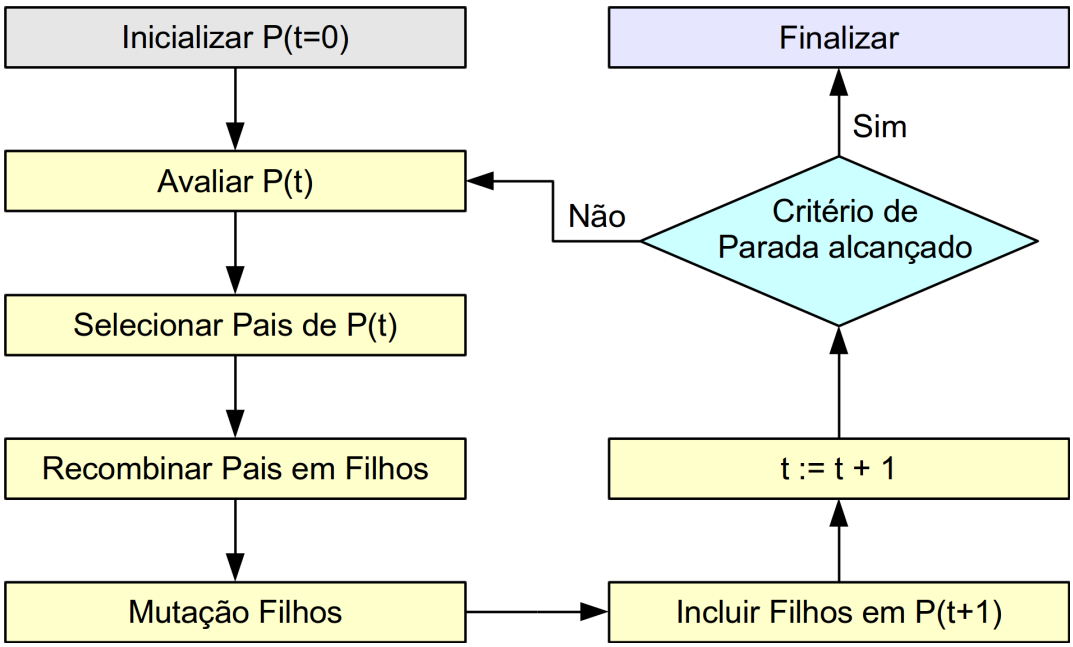


Figura 3.1: Fluxograma de um AG (adaptado de (CORDÓN et al., 2004)).

3.2 Sistemas *Fuzzy* Genéticos

As técnicas de computação flexível têm alcançado grande sucesso na solução de problemas práticos nos últimos 30 anos. Dentre as principais técnicas desta linha de pesquisa encontram-se as redes neurais artificiais, os algoritmos baseados em inteligência de enxames, os algoritmos evolucionários e a lógica *fuzzy*. Abordagens híbridas que visam aproveitar o melhor de cada técnica vem atraindo atenção considerável na comunidade científica já há algum tempo. Uma das abordagens mais populares é a hibridização entre a lógica *fuzzy* e os algoritmos genéticos originando os *Sistemas Fuzzy Genéticos* (SFG) (CORDÓN et al., 2004; HERRERA, 2008).

Os SFG são basicamente sistemas *fuzzy* acoplados a um processo de aprendizado baseado em computação evolucionária, incluindo AG, programação genética e estratégias evolucionárias, entre outras (HERRERA, 2008).

Os SFBR são uma das mais importantes aplicações da TCF e são o tipo de sistema *fuzzy* de interesse nesta proposta. A abordagem mais natural no projeto de SFBR baseia-se na elicitação do conhecimento de um especialista humano e na codificação desse conhecimento na BC do SFBR. Entretanto, esta abordagem tem uma série de limitações já apresentadas nesta proposta (ver Capítulo 1). A definição automática de SFBR a partir de dados, por meio de AG, é uma das áreas mais bem sucedidas ao longo do tempo, como pode ser evidenciado na literatura (CORDÓN et al., 2004; HERRERA, 2008; ISHIBUCHI; NAKASHIMA; NOJIMA, 2010; FAZZOLARI et al., 2013; FERNÁNDEZ et al., 2015) e tenta prover uma solução a estas limitações.

As capacidades de busca e otimização dos AG podem ser exploradas neste contexto de maneira bastante abrangente, possibilitando seu uso na definição ou melhoria de vários componentes dos SFBR, tais como a BD e sua granularidade, a BR ou mesmo a BC inteira. Além disso, devido à facilidade em codificar o problema que os AG oferecem, estes permanecem como uma das poucas técnicas capazes de permitir que os especialistas das áreas de domínio interfiram no processo, definindo quais componentes evoluir (HERRERA, 2008).

As abordagens para SFG subdividem-se basicamente em torno de dois processos, otimização e aprendizado, além de levarem em conta a existência prévia de elementos da BC. Em Fernández et al. (2015) encontra-se uma taxonomia completa para SFG. Em linhas gerais, ela pode ser resumida como:

- **Aprendizado Genético:** Busca evoluir ou construir componentes da BC em função de outros componentes do SFBR. As principais abordagens nesta linha são:

  - **Aprendizado de regras:** Consiste na construção da BR por meio de dados numéricos utilizando uma BD pré-definida, geralmente por meio de partições uniformes (THRIFT, 1991);
  - **Seleção de regras:** Consiste em melhorar a performance do SFBR eliminando da BR as regras redundantes, conflitantes ou irrelevantes, visto que, geralmente, quando uma BR é extraída por algum método de aprendizado de máquina, o resultado é uma BR com uma quantidade grande de regras, o que torna difícil a interpretação do SFBR (ISHIBUCHI et al., 1995);
  - **Aprendizado da BD:** Consiste em uma abordagem que gera a BD adaptando a forma e o número de CF. Pode utilizar um processo de geração da BD a priori, para posterior geração da BR, ou um processo de geração embutida da BR, para cada BD encontrada (CORDÓN; HERRERA; VILLAR, 2001);
  - **Aprendizado simultâneo dos componentes da BC:** Consiste nas abordagens que tentam derivar de dados numéricos tanto a BD quanto a BR simultaneamente. Apesar de estas abordagens poderem oferecer definições melhores, elas tendem a esbarrar em um espaço de busca muito vasto, comprometendo o desempenho (HOMAIFAR; MCCORMICK, 1995).

- **Otimização Genética:** Busca melhorar a performance do SFBR, por meio da otimização da BD previamente definida ou dos parâmetros do mecanismo de inferência, partindo da BR já definida. As principais abordagens nesta linha são:

  - **Otimização dos parâmetros da BD:** Utiliza a BC como um todo para ajustar apenas a forma das funções de pertinência dos CF na BD, mantendo o número de termos linguísticos intacto (CASILLAS et al., 2005);
  - **Adaptação genética do MI:** Utiliza as expressões parametrizadas no MI para alcançar maior cooperação com as regras *fuzzy*, porém sem perder a interpretabilidade linguística da regra (ALCALÁ-FDEZ et al., 2007);
  - **Adaptação genética dos métodos de defuzzificação:** Consiste em aplicar um AG para ajustar os parâmetros de uma função de média ponderada, usada para calcular o valor final do método de defuzzificação, a partir dos valores resultantes da defuzificação de cada CF inferido pelas regras (KIM; CHOI; LEE, 2002).

Para a abordagem que será proposta, uma utilização de AG será no aprendizado da BR, visto este ser o componente mais oneroso de ser descrito pelos especialistas, principalmente quando

o número de variáveis é grande, aumentando exponencialmente o número de regras possíveis. Neste contexto as abordagens de codificação do AG para aprendizado da BR são inúmeras, entretanto as mais usuais são (STAVRAKOUDIS; THEOCHARIS; ZALIDIS, 2010):

- **Abordagem Pittsburg:** Cada cromossomo da população codifica uma BR inteira e elas competem entre si através do processo evolucionário. Sua principal desvantagem é o crescimento demasiado do cromossomo quando o problema tem muitas variáveis;
- **Abordagem Michigan:** Cada cromossomo codifica uma única regra e a população inteira forma a BR. Também é conhecida como sistema classificador, pois as regras competem entre si no processo evolucionário. Não possui a desvantagem da abordagem anterior, entretanto deve-se ser criterioso na sua aplicação, visto que as regras da BR devem cooperar ao invés de competir;
- **Abordagem cooperativa-competitiva:** A população ou uma parcela dela codificam a BR, entretanto os cromossomos competem e cooperam simultaneamente;
- **Abordagem Iterativa:** Cada cromossomo codifica uma única regra, entretanto a base de regras é formada iterativamente. A cada geração do AG, o cromossomo (regra) mais apto da população é adicionado à BR, sendo este processo repetido até completar-se a mesma. Uma das considerações sobre esta abordagem é que apesar de ela reduzir o espaço de busca a cada iteração, ela pode levar a BR com regras conflitantes, uma vez que a seleção não leva em conta as regras que já saíram da população.

3.2.1 Sistemas *Fuzzy* Genéticos Multiobjetivo

As abordagens pioneiras, baseadas em AG de único objetivo, tiveram e ainda têm um papel de destaque na área dos SFG, entretanto, geralmente baseiam-se na acurácia como o principal critério de avaliação do desempenho, ou definem uma única função objetivo que combina parâmetros ponderados que representam critérios conflitantes. Muito tem sido discutido sobre a baixa interpretabilidade dos SFBR gerados de maneira automatizada por meio de dados, uma vez que a interpretabilidade é uma das principais vantagens e critério de escolha da técnica de SFBR (ANTONELLI et al., 2010).

Obter altos níveis de interpretabilidade e acurácia, ao mesmo tempo, são objetivos conflitantes, visto que na prática uma característica influencia negativamente a outra. Uma tendência na área é a preocupação no balanceamento entre estes dois aspectos (CORDÓN et al., 2004; HERRERA, 2008; ISHIBUCHI, 2007). Particularmente, quando se está modelando fenômenos

ou sistemas do mundo real, este balanceamento é bastante desejado pois é importante entender o funcionamento do modelo.

Nos últimos anos, têm surgido algumas abordagens que se propõem a contornar este problema. As mais promissoras são as baseadas em AGMO em que dois objetivos são definidos para a geração do sistema: maximizar a acurácia e maximizar a interpretabilidade. Neste tipo de abordagem, o algoritmo fornece um conjunto de soluções não dominadas[1] em uma simples execução, formando uma curva de SFBR possíveis, variando a compensação entre erro e complexidade (Figura 3.2). Esta curva busca sempre o *Ótimo de Pareto*, ou seja, cada elemento no conjunto de soluções maximiza os objetivos para um determinado nível de compensação. O limite formado por todos os níveis é denominado *Fronteira de Pareto*. Estas abordagens são conhecidas atualmente como *Sistemas Fuzzy Genéticos Multiobjetivo* (SFGMO) (ISHIBUCHI; NAKASHIMA; NOJIMA, 2010).

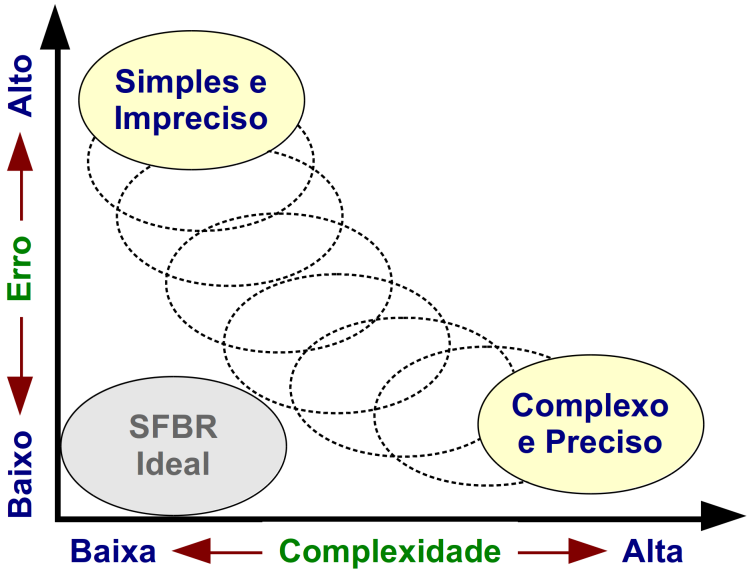


Figura 3.2: SFBR não dominados ao longo da curva Complexidade X Erro (adaptado de (ISHIBUCHI; NAKASHIMA; NOJIMA, 2010)).

Uma das vantagens dos SFGMO sobre os de objetivo simples, além de deixar a busca pelo balanceamento entre os objetivos conflitantes a cargo do processo evolutivo, é a possibilidade oferecida ao especialista da área de domínio de escolher a solução mais adequada ao contexto de um conjunto de SFBR gerados, visto que complexidade é dependente do problema sendo abordado (ISHIBUCHI; NAKASHIMA; NOJIMA, 2010; ISHIBUCHI, 2007). A incorporação das preferências do especialista no processo de projeto de SFBR é uma tendência na comunidade científica da área de AGMO (ISHIBUCHI, 2007).

Embora ainda não exista um consenso, a complexidade de SFBR geralmente é medida

[1] Solução não dominada é aquela em que não é possível melhorar uma variável sem piorar outra.

por características da BC, como o número de regras e quantidade de antecedentes das regras na BR, granularidade das partições *fuzzy* e separabilidade das funções de pertinência na BD (ANTONELLI et al., 2010).

As abordagens mais recentes, nomeadas de *Segunda Geração*, focam na otimização destes elementos e em simultaneamente aprender e/ou otimizar a BD e a BR como em (ALCALÁ et al., 2010; ANTONELLI et al., 2010, 2011; CANNONE; ALONSO; MAGDALENA, 2011).

Os algoritmos nos quais baseiam-se grande parte das abordagens bem sucedidas citadas na literatura (ISHIBUCHI, 2007; ALCALÁ et al., 2009; ISHIBUCHI; NAKASHIMA; NOJIMA, 2010; SHUKLA; TRIPATHI, 2012a) são o MOEA/D (ZHANG; LI, 2007), NSGA-II (DEB et al., 2002), MOGA (FONSECA; FLEMING, 1993), SPEA2 (ZITZLER; LAUMANNS; THIELE, 2001), PAES (KNOWLES; CORNE, 2000) e seus variantes.

Capítulo 4

Interpretabilidade na Construção de Sistemas *Fuzzy* Baseados em Regras

Durante o processo de construção de SFBR linguísticos (Mamdani), uma necessidade é a avaliação da performance do sistema segundo métricas que reflitam os aspectos desejados para o produto final do processo. Os dois principais aspectos a serem levados em conta são a acurácia, que é a capacidade de representar fielmente o comportamento do sistema real, e a interpretabilidade, que é a capacidade de expressar o comportamento do sistema real de uma maneira compreensível.

Embora as medidas de precisão sejam simples e bem conhecidas, as medidas de interpretabilidade são difíceis de definir, pois este conceito é abstrato e depende de quem está avaliando-o. Embora não haja um consenso, a maioria dos pesquisadores concorda que tal medida deve levar em conta, principalmente, fatores acerca da estrutura do modelo. A escolha de medidas de interpretabilidade apropriadas ainda é um problema em aberto (ALONSO et al., 2011).

Os dois principais tipos de abordagens para o tratamento da interpretabilidade em SFBR lingúisticos são (GACTO; ALCALÁ; HERRERA, 2010):

- **Baseadas na complexidade:** Buscam diminuir a complexidade estrutural do modelo obtido, geralmente medida pelo número de regras, número de premissas nas regras, número de variáveis, número de conjuntos *fuzzy*, etc.;
- **Baseadas na semântica:** Buscam preservar a integridade semântica dos conjuntos *fuzzy*, geralmente impondo restrições sobre as funções de pertinência ou avaliando medidas como cobertura, distinguibilidade, etc.

Segundo Alonso et al. (2011), onde pode ser encontrada uma taxonomia para medidas de

interpretabilidade em SFBR linguísticos, estes aspectos devem ser considerados em ambos os componentes da BC. Sendo assim, o autor propõe uma classificação em quadrantes que pode ser vista na Tabela 4.1. Cada medida será detalhada nas seções subsequentes.

Tabela 4.1: Taxonomia da interpretabilidade de SFBR (adaptada de (ALONSO et al., 2011)).

	Base de Regras	Base de Dados
Complexidade	Q_1 Número de regras Número de condições (no antecedente)	Q_2 Número de CF Número de variáveis
Semântica	Q_3 Consistência das regras Regras disparadas em paralelo Transparência da estrutura das regras (pesos, etc.) Cointenção	Q_4 Completude ou cobertura Normalização Distinguibilidade Complementaridade Medidas relativas

4.1 Base de regras

4.1.1 Complexidade

Para manter sob controle a complexidade na base de regras, segundo Alonso et al. (2011), há um consenso e um grande número de trabalhos que utilizam como métricas:

- **Número de regras:** Segundo o princípio da navalha de Occam (o melhor modelo é o mais simples possível que seja capaz de se adequar ao comportamento do sistema) (MITCHELL, 1997), o conjunto de regras deve ser o menor possível capaz de manter a performance desejada.
- **Número de condições:** O número de condições no antecedente das regras não deve exceder o limite de 7 ± 2. Além disso o número de condições deve ser o menor possível, facilitando a leitura da regra e favorecendo a generalização do modelo, mantendo a performance desejada.

Algumas considerações devem ser levadas em conta quando da aplicação destas medidas:

- Diferenças na simplicidade somente são relevantes quando são grandes o bastante, por

exemplo uma BR contendo 30 regras e uma outra com 32 regras estão na prática no mesmo nível de complexidade.

- De acordo com Zhou e Gan (2008), o sistema deve ser simplificado sem que isso afete sua usabilidade, ou seja, ele deve ser tão simples quanto for suficiente para que ofereça desempenho aceitável na solução do problema. O papel de determinar qual é este ponto de equilíbrio, geralmente, é desempenhado por um especialista de domínio de conhecimento.

4.1.2 Semântica

Assumindo que as partições *fuzzy* são interpretáveis no nível semântico, algumas medidas utilizadas na literatura para controlar a interpretabilidade semântica a nível de BR são:

- **Consistência da BR:** O que significa a ausência de regras contraditórias na BR, no sentido de que regras com condições similares no antecedente devem ter consequentes similares. A análise de consistência pode ser diferente dependendo do tipo de problema (classificação, regressão, controle, etc.).
- **Regras disparadas em paralelo:** Consiste em minimizar o número de regras ativadas ao mesmo tempo durante a inferência. Através do controle desta medida, é possível preservar os significados individuais das regras linguísticas que compõem a BR.

Para manter a interpretabilidade no que tange à semântica da base de regras, algumas propriedades estruturais nas regras devem ser consideradas para garantir a consistência. Este aspecto será abordado, de maneira contextualizada, na seção 5.5.

4.2 Base de dados

4.2.1 Complexidade

Para manter a complexidade da BD sob controle alguns critérios devem ser levados em consideração (ALONSO; MAGDALENA; GONZÁLEZ-RODRÍGUEZ, 2009; ALONSO et al., 2011; GUILLAUME; CHARNOMORDIC, 2012); tais como:

- **Número de variáveis:** A redução do número de variáveis pode melhorar a interpretabilidade da base de conhecimento de maneira exponencial (maldição da dimensionalidade).

- **Quantidade de conjuntos:** O número de conjuntos *fuzzy* na partição da variável deve ser adequado, não deve exceder o limite de 7 ± 2, que é a quantidade de conceitos distintos que os seres humanos são capazes de gerenciar (MILLER, 1956). À medida que a quantidade de CF cresce, a acurácia pode crescer também, mas a relevância do modelo irá decrescer.

Deve-se observar que tanto a quantidade de variáveis quanto suas respectivas granularidades no particionamento *fuzzy* determinam a especificidade ou generalidade de um SFBR e têm influência direta no tamanho da BR e, consequentemente, em sua complexidade também.

4.2.2 Semântica

Processos de definição do particionamento *fuzzy* das variáveis por métodos que visem precisão (aprendizado de máquina, estatísticos, etc.), podem resultar em partições *fuzzy* complexas e de difícil interpretação pelo especialista. Pode ser visto na figura 4.1 um exemplo de particionamento *fuzzy* complexo (com grande sobreposição entre os CF), o que deteriora a interpretabilidade dos conceitos representados na partição. A maior parte das abordagens para resolver esse problema impõe restrições sobre as funções de pertinência dos CF na partição. As medidas mais frequentemente utilizadas para isso são:

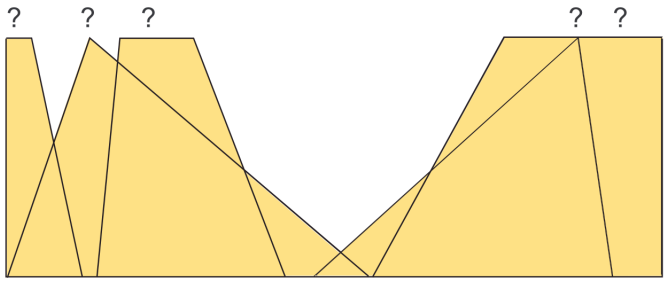


Figura 4.1: Exemplo de partição *fuzzy* com semântica pobre (ALONSO et al., 2011).

- **Distinguibilidade:** Cada conjunto *fuzzy* deve representar um conceito diferente por meio de sua função de pertinência, garantindo a integridade semântica;
- **Cobertura:** Cada valor no UD da variável deve pertencer significativamente a pelo menos um conjunto *fuzzy*.
- **Normalização:** Todos os conjuntos *fuzzy* devem ser normais (Altura = 1);
- **Sobreposição:** Todos os conjuntos *fuzzy* devem se sobrepor significativamente, de maneira que permita uma transição suave entre os conceitos definidos para a variável.

- **Complementaridade:** Para cada valor no UD da variável, a soma de todos os graus de pertinência dos conjuntos *fuzzy* na partição deve ser próxima de 1. Isso favorece uma distribuição uniforme dos conceitos na partição.

4.3 Modelos de particionamento

4.3.1 Partições *fuzzy* fortes

Um modelo de particionamento capaz de cumprir todas estas restrições apresentadas na Seção 4.2.2, são as *Partições Fuzzy Fortes* (PFF). Um exemplo deste tipo de partição usando conjuntos com formato triangular foi mostrado na Figura 2.9. Neste tipo de particionamento, o somatório dos graus de pertinência ($F_i(x)$) de qualquer valor no UD da variável resulta em 1, como na equação (4.1),

$$\begin{cases} \forall x & \sum\limits_{i=1,2,\ldots,m} & F_i(x) = 1 \\ \forall F_i & \exists x & F_i(x) = 1 \end{cases} \tag{4.1}$$

em que m é o número de conjuntos *fuzzy* e $F_i(x)$ é o grau de pertinência do F-ésimo conjunto *fuzzy* na partição.

Nesta proposta, durante a etapa de definição das partições iniciais das variáveis que irão compor a BD do SFBR, serão utilizadas PFF, compostas por conjuntos *fuzzy* definidos por funções de pertinência triangulares e/ou trapezoidais. Estas partições são consideradas como uma das mais fáceis para o especialista entender e associar conceitos.

4.3.2 Partições *fuzzy* transparentes

A última etapa do processo proposto no Capítulo 5, que consistirá em uma otimização das partições das variáveis da BD do SFBR, poderá relaxar um pouco as restrições das PFF, utilizando partições *fuzzy* transparentes. Este modelo de particionamento visa oferecer flexibilidade suficiente para que seja possível melhorar a acurácia de um SFBR, através da otimização dos conjuntos *fuzzy* que compõem sua BD, sem prejudicar a integridade semântica da partição, e, portanto, sua interpretabilidade (PULKKINEN; KOIVISTO, 2010).

Para uma partição *fuzzy* ser transparente (Figura 4.2), além dos critérios de legibilidade já apresentados na Seção 4.2.2, algumas restrições devem ser atendidas (OLIVEIRA, 1999):

- **Simetria:** A forma de todos os CF deve ser simétrica.
- **Condição α:** Em qualquer ponto de interseção de dois CF, o grau de pertinência desse ponto deve ser menor ou igual a α. Este parâmetro assegura que os CF não se sobreponham de maneira excessiva, ao ponto de comprometer a semântica.
- **Condição γ:** No centro de cada CF, nenhum outro CF pode ter um grau de pertinência maior ou igual a γ. Este parâmetro assegura a distinguibilidade dos CF da partição.
- **Condição β:** Em cada ponto do UD da variável, pelo menos um CF deve ter um grau de pertinência maior ou igual a β. Este parâmetro assegura a boa cobertura da partição.

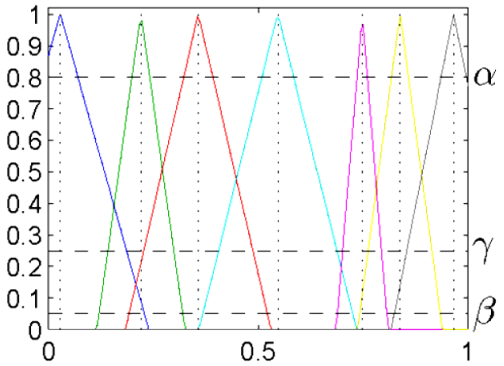


Figura 4.2: Exemplo de partição *fuzzy* transparente usando conjuntos triangulares (PULKKINEN; KOIVISTO, 2010).

Capítulo 5

ABORDAGEM PROPOSTA

Nesta abordagem o contexto principal para o qual nos direcionamos são problemas em que a interação humana é necessária, como por exemplo, nos sistemas de suporte à decisão. Em tais sistemas, o número de variáveis envolvidas não deve ultrapassar o limite do entendimento humano, pois, o especialista deve ser capaz de interpretar as inferências realizadas pelo modelo. Não é o foco da abordagem tratar problemas de controle, onde certamente já existem técnicas bem consolidadas disponíveis (porém não há restrição ao uso, visto que em alguns casos, a capacidade de interpretação é desejável, como por exemplo, na área de robótica).

A abordagem parte do princípio de que uma das principais razões para a utilização de SFBR é quando a semântica é necessária. A grande maioria dos trabalhos propostos na área de construção automatizada de SFBR, na última década, se preocupa que o resultado do processo seja interpretável, entretanto, mesmo que o sistema seja legível, ele pode não ter significado algum para o especialista. Isto pode ocorrer pois, em sistemas complexos, irão frequentemente existir inúmeras soluções igualmente boas para o problema, mas, o especialista espera sempre enxergar no modelo elementos que se assemelhem à sua experiência na área de conhecimento sendo tratada. Partimos então do pressuposto que a melhor maneira de resolver este problema é envolvendo diretamente o especialista na modelagem, levando adiante o aprendizado automático partindo dos dados, seguindo as restrições tomadas pelo especialista, garantindo assim a semântica do produto final.

Uma característica inerente a uma amostra de dados é sua incompletude. Qualquer tentativa de assumir que os dados representam fielmente o mundo real deve ser cuidadosa. Uma amostra de dados não é capaz de cobrir todas as situações possíveis, sobretudo em sistemas complexos (GUILLAUME; CHARNOMORDIC, 2012). Além disso, dados estão suscetíveis a erros das mais diversas naturezas (*outliers*), como erros na coleta, transmissão, armazenamento, formatação, erros em sensores, valores faltantes, entre outros. A Figura 5.1 ilustra esse problema;

podem existir exemplos corretos fora do escopo dos dados, bem como, exemplos errados dentro do escopo da amostra de dados.

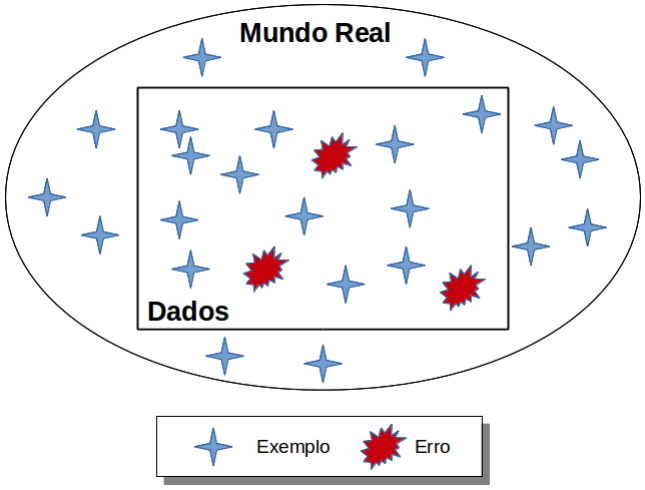


Figura 5.1: Ilustração das características inerentes aos dados.

Técnicas de modelagem clássicas, como as técnicas estatísticas e de aprendizado de máquina, dependem extremamente de uma quantidade de dados suficiente para que sejam eficazes, além de serem influenciadas diretamente pela qualidade dos dados. Uma abordagem para lidar com este problema é utilizar o especialista para preencher as lacunas no conhecimento onde não há dados ou os dados não são confiáveis.

Por outro lado, um especialista geralmente também não é capaz de explicar um sistema em seus mínimos detalhes. Seu conhecimento tem um alto nível de generalização e a formalização deste conhecimento, geralmente, é uma tarefa complexa para o especialista (JANSSEN et al., 2010; GUILLAUME; MAGDALENA, 2006). Uma metodologia que facilite a extração de conhecimento do especialista é bastante desejável.

A questão chave nesta abordagem é integrar estes dois tipos diferentes de conhecimento, pois eles são de fato complementares no que tange à modelagem. Esta integração se dará deixando o especialista no nível linguístico, e usando os dados em nível numérico para viabilizar o aprendizado e otimizar o modelo.

A Figura 5.2 ilustra em linhas gerais o processo proposto. Ele é composto de etapas sequenciais, cada uma fornecendo subprodutos para a etapa seguinte. Em todas elas há a integração entre o conhecimento que pode ser extraído de dados e do especialista, que está no comando das decisões. Trata-se de um processo iterativo e interativo, onde o especialista pode retornar e repetir os passos que julgar necessários para obter os resultados desejados. A ferramenta de software desenvolvida para implementar a abordagem é apresentada no Capítulo 6.

É importante ressaltar, também, que a abordagem foi estruturada de maneira que o especi-

alista não é "obrigado" em nenhum momento a fornecer informação onde ele não se sentir apto para isso, deixando a obtenção dos parâmetros restantes a cargo de aprendizado de máquina. Isso garante flexibilidade ao processo e melhora a qualidade do conhecimento obtido do especialista. Outra consequência desta heurística, no que tange à ferramenta de software, é que se um especialista não estiver disponível, um SFBR pode ser construído utilizando somente os dados (de maneira equivalente a outras abordagens com esta finalidade).

Em todas as etapas do processo, os SFBR que estaremos tratando utilizarão o modelo de inferência de Mamdani (inferência max-min, conjunção pelo mínimo e agregação pelo máximo). Para saída do sistema, no caso de problemas de regressão, será adotada a defuzzificação pelo centro de gravidade; enquanto para problemas de classificação, o raciocínio *fuzzy* clássico (regra vencedora) (CHI; YAN; PHAM, 1996) será utilizado.

Como elemento principal das etapas de aprendizado e otimização dos componentes do SFBR, foi adotado um AGMO. O algoritmo original, seus operadores e funções objetivo foram customizados, visando: o estreitamento do espaço de busca; a obtenção de soluções compactas e a obtenção de uma fronteira de Pareto mais diversificada; bem como, a incorporação das preferências do especialista. Apresentaremos estas adaptações durante a descrição de cada etapa.

O primeiro passo do processo é a definição do domínio do problema e da sua natureza (classificação ou regressão). Isto irá determinar uma série de parâmetros que serão adotados nas etapas subsequentes. Para representar essa definição, o especialista deve fornecer um conjunto de dados (amostra) representativo do problema sendo abordado. Se o atributo meta deste conjunto for definido como nominal, será considerado então um problema de classificação, analogamente, se este atributo for numérico, será considerado como regressão. A seguir examinaremos de maneira mais detalhada cada uma das etapas seguintes.

5.1 Seleção de atributos

A Seleção de Atributos é uma etapa de pré-processamento presente na maioria dos processos que envolvem aprendizado de máquina. Como o próprio nome já diz, o objetivo é escolher um subconjunto de atributos (variáveis) que ofereça resultado equivalente ao conjunto original, quando submetidos a algum algoritmo.

Embora esta etapa não seja incluída em grande parte das abordagens semelhantes à proposta, ela pode oferecer grande vantagem quando adotada. Em primeira instância, a seleção de atributos gera modelos resultantes mais simples, o que, segundo o princípio lógico da *navalha*

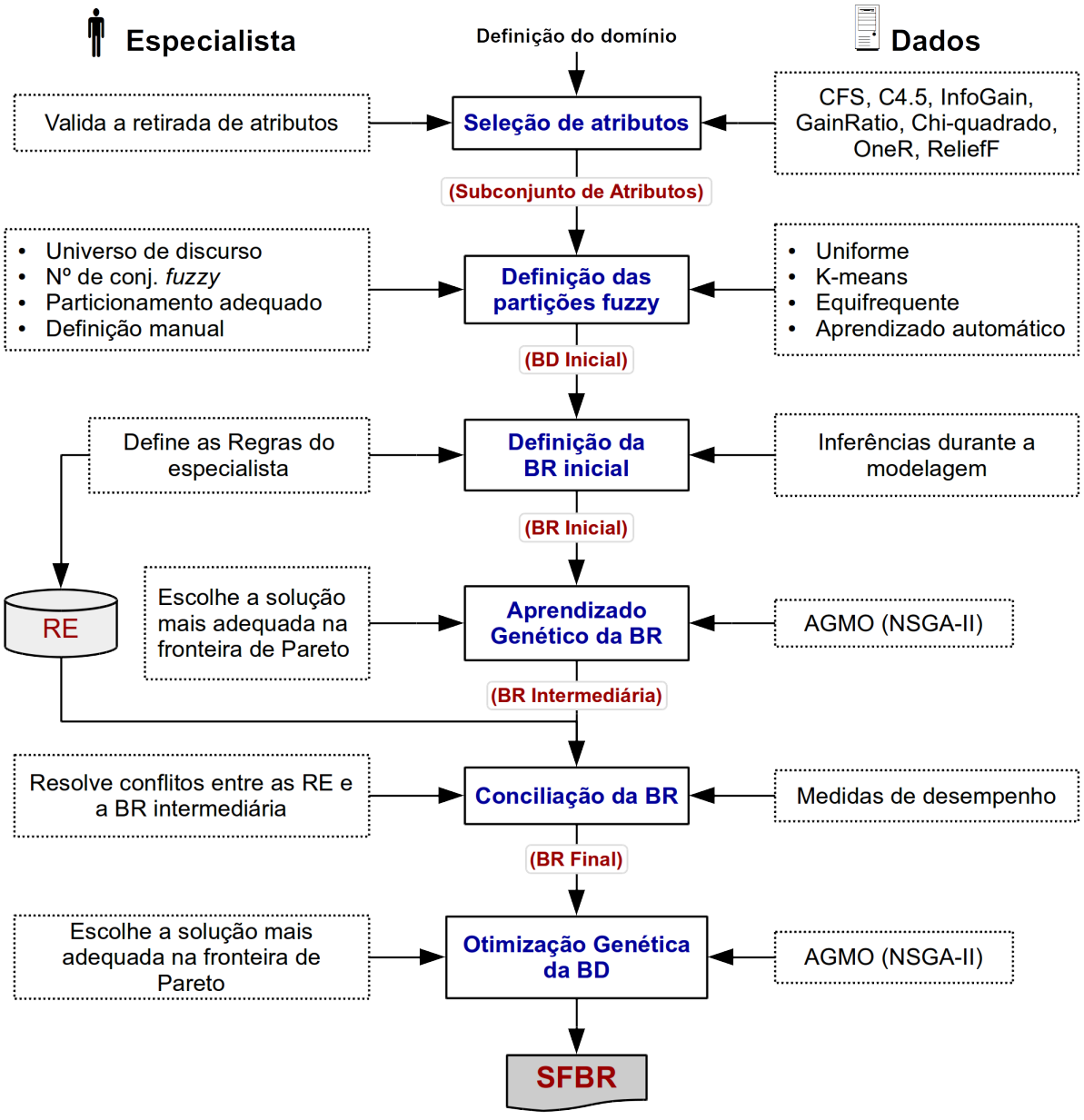


Figura 5.2: Visão geral da abordagem proposta para integração do conhecimento do especialista e e extraído de dados, à construção de um SFBR.

de Occam (MITCHELL, 1997), são frequentemente os mais adequados. Em segunda instância, quanto menos variáveis o modelo tiver, menor será sua complexidade, e consequentemente, mais interpretável ele será para o usuário/especialista. Por último, a técnica utilizada para aprendizado da BR e otimização da BD (AGMO) é computacionalmente custosa e a quantidade de variáveis tem influência direta no tamanho do espaço de busca do problema. A cada variável adicional considerada no modelo, o espaço de busca cresce exponencialmente, demandando um número muito maior de iterações para que uma solução adequada seja encontrada, ou mesmo inviabilizando todo o processo. Ainda a nível de algoritmo, é possível que algumas variáveis acabem adicionando ruído ao invés de contribuir para a predição do atributo meta.

Para selecionar um bom subconjunto de atributos, existem basicamente duas abordagens. A primeira é fazer uma avaliação de maneira independente do algoritmo que será adotado, baseando-se em características intrínsecas dos dados; a segunda é avaliar a performance do subconjunto usando o próprio algoritmo que será empregado posteriormente. A primeira é chamada abordagem *Filtro*, pois tenta filtrar os atributos irrelevantes, produzindo um subconjunto mais promissor antes do processo de aprendizado de máquina. A segunda é chamada abordagem *Wrapper*, pois o algoritmo de aprendizado está envolto (*wrapped*) no processo de seleção (WITTEN; FRANK; HALL, 2011).

A abordagem *Wrapper* (KOHAVI; JOHN, 1997) geralmente oferece excelentes resultados, mas, devido ao já mencionado alto custo computacional dos AGMO, é inviável adotá-la nesta proposta.

Embora muitos métodos tenham sido propostos para efetuar a avaliação do subconjunto de atributos de maneira independente do algoritmo (Filtro), não existe um que seja universalmente aceito. Para minimizar este efeito, adotaremos alguns métodos do tipo filtro de maneira conjunta, ao invés de escolher apenas um método para realizar a seleção.

Ainda acerca das abordagens Filtro, existem dois tipos de métodos: os que avaliam a relevância de cada atributo em relação ao atributo meta e retornam como resultado um ranqueamento dos atributos, onde o usuário escolhe um ponto de corte para o descarte de atributos; e os que avaliam a relevância dos atributos de maneira conjunta, retornam o subconjunto mais promissor (GUYON; ELISSEEFF, 2003). Ambos os tipos oferecem vantagens na abordagem proposta: o primeiro, possibilita que o especialista escolha de maneira flexível os atributos que deseja descartar, orientando-se pelo ranking; e o segundo considera que dois ou mais atributos, em conjunto, podem ter a capacidade de prever o atributo meta, que não o teriam quando avaliados de maneira isolada.

Assim como em qualquer processo de aprendizado de máquina, é importante na seleção de

atributos, que o resultado da aplicação dos algoritmos seja validado. Existem várias maneiras de realizar esta avaliação, dentre elas, uma das mais bem aceitas é a validação cruzada de *N* partições (*N-fold cross-validation*) (STONE, 1974).

5.1.1 Processo para seleção de atributos

Nesta etapa o especialista é encarregado de escolher quais atributos devem ser mantidos para as etapas subsequentes da abordagem. Como subsídio para essa escolha, serão aplicados ao conjunto de dados, tanto algoritmos que retornam um ranking de atributos, quanto algoritmos que retornam um subconjunto de atributos. O resultado desta análise será apresentado ao usuário no formato de tabela (ver exemplo na Tabela 5.1), onde constarão, para cada atributo:

- **Ranking médio do atributo:** Calculado pela média (e respectivo desvio padrão) do ranking do atributo em cada algoritmo que retorna um ranking;
- **Seleção em subconjunto:** Mostra, para cada algoritmo que retorna um subconjunto, se o atributo foi (ou não) selecionado no subconjunto de resultado do algoritmo;
- **Recomendação:** Uma heurística é usada para mostrar uma recomendação ao especialista sobre o descarte do atributo em questão (abordada mais adiante).

Para a validação da aplicação dos algoritmos será adotada validação cruzada de 10 partições (estratificadas por classe, no caso de um problema de classificação). Este processo consiste em aplicar o algoritmo em cada partição dos dados, retornando: a média do ranking que os atributos alcançaram em cada partição; ou, no caso dos algoritmos que retornam um subconjunto, os atributos que foram selecionados em pelo menos 50% das partições.

Os algoritmos selecionados para este processo são:

- **C4.5:** É um algoritmo de indução de árvores de decisão que utiliza a medida de taxa de ganho de informação (QUINLAN, 1993). Este algoritmo tem como viés indutivo obter a árvore mais rasa possível (composta por menos nós/variáveis). Portanto, ele efetua uma seleção de atributos embutida no próprio algoritmo. O subconjunto dos atributos que aparecerem em algum nó na árvore gerada é considerado o subconjunto de atributos selecionado;
- **CFS:** Avalia a relevância de um subconjunto de atributos considerando a capacidade preditiva de cada atributo em relação à classe e a redundância entre eles. Retorna o subcon-

junto de atributos com a maior correlação com a classe e a menor correlação entre eles mesmos (HALL, 1999).

- **InfoGain e GainRatio:** Ambos utilizam como métrica a entropia para avaliar quais atributos melhor dividem o conjunto de dados em relação à classe (WITTEN; FRANK; HALL, 2011). A diferença entre eles é que o GainRatio leva em conta no cálculo a proporção da classe nos dados. Retorna um ranking de atributos.
- **Chi-quadrado (χ^2):** Avalia a relevância de cada atributo em relação à classe por meio da estatística do Chi-quadrado (LIU; SETIONO, 1995). Retorna um ranking de atributos.
- **OneR:** Avalia a capacidade preditiva de cada atributo em relação à classe construindo regras com apenas uma variável e avaliando a acurácia destas regras no conjunto de dados. Pode ser visto como uma árvore de decisão de apenas um nível (HOLTE, 1993). Retorna um ranking dos atributos cujas regras obtiveram melhor desempenho.
- **ReliefF:** Utiliza aprendizado baseado em instância para atribuir um peso para cada atributo. Este peso reflete a habilidade de distinguir os valores do atributo meta (KIRA; RENDELL, 1992). Retorna um ranking de atributos.

Em problemas de regressão nem todos os métodos são aplicáveis (apenas o CFS e o ReliefF), valendo ressaltar, também, que a seleção de atributos deve ser mais cautelosa, visto que a relação de uma variável com o atributo meta é difícil de ser avaliada. Geralmente a saída é produto da influência em maior ou menor grau de todas ou de parte das variáveis.

Após a aplicação de todos os algoritmos, baseando-se nos resultados e na quantidade de atributos no conjunto de dados, a seguinte heurística é usada para recomendar sobre a retirada ou permanência de um atributo:

- Manter todos os atributos selecionados pelo CFS ou C4.5;
- Se o conjunto de dados possui menos de 10 atributos, manter todos eles;
- Se o conjunto de dados possui menos de 20 atributos, manter os 70% melhor ranqueados;
- Se o conjunto de dados possui menos de 30 atributos, manter os 50% melhor ranqueados;
- Se o conjunto de dados possui mais de 30 atributos, manter os 15 melhor ranqueados e descartar o restante dos atributos.

O produto final desta etapa é um subconjunto de atributos de interesse no processo de modelagem.

Tabela 5.1: Exemplo de tabela apresentada ao especialista após aplicação do processo para seleção de atributos sobre o conjunto de dados *Diabetes* disponível no repositório da UCI (UCI, 2015).

Recomendacao	Ranking(stdev)	Sel.	# Atributo	
Manter	1.1 +– 1.98	* t	2 plas	
Manter	2.3 +– 1.43	* t	6 mass	
Manter	3.2 +– 1.65	* t	8 age	
Manter	4.8 +– 1.64		1 preg	
Manter	5.1 +– 1.57		5 insu	
Manter	5.8 +– 2.17		4 skin	50%
Manter	6.1 +– 2.1	* t	7 pedi	
Manter	7.6 +– 1.77	t	3 pres	70%

* = CFS t = C4.5

5.2 Definição das partições *fuzzy*

O objetivo desta etapa é subsidiar o especialista na definição das partições *fuzzy* de cada variável selecionada. Além da possibilidade de definição manual das partições, para cada variável, o conjunto de dados será utilizado para oferecer ao especialista algumas opções de particionamento automático, ou em última instância, utilizar aprendizado de máquina para definir as partições.

A análise exploratória de dados é uma ferramenta interessante para entender qualquer problema. Para cada variável, serão apresentados ao especialista: um gráfico de histograma, para que ele possa enxergar a distribuição dos dados ao longo do UD da variável; e um diagrama de caixa (*boxplot*), que permitirá ao especialista, através da análise da distância interquartílica, visualizar possíveis valores extremos (*outliers*), removê-los do conjunto de dados ou, caso seja de interesse, definir um conjunto *fuzzy* para eles. Além dos gráficos, são mostradas algumas métricas de estatística descritiva para cada variável, tais como: média, desvio padrão, mínimo, máximo, tipo (numérico ou nominal), quantidade de valores distintos, percentual de valores ausentes, percentual de valores únicos (que só aparecem uma vez no conjunto de dados), percentual de valores inteiros e percentual de valores com porção fracionária.

Especificamente em problemas de classificação, através do histograma da classe, o especialista também poderá identificar algum problema de desbalanceamento de classes na amostra de dados fornecida.

O modelo de particionamento adotado nesta etapa é o de PFF com funções de pertinên-

cia triangulares (ou trapezoidais, nos extremos da variável). No caso das variáveis nominais (como a classe em problemas de classificação), é adotada uma partição com conjuntos do tipo *singleton*, onde cada valor distinto é representado por um conjunto.

Observando os indicadores apresentados e usando seu conhecimento acerca do domínio do problema, o especialista deve, então, definir a maior quantidade possível de parâmetros das partições, a fim de que a semântica esperada por ele seja mantida na BR, que é uma das condições para que o especialista possa interpretar as inferências feitas pelo modelo resultante. Entretanto, nenhum dos parâmetros é mandatório, podendo ser assinalados como *'desconhecido'* e definidos por meio algorítmico mais tarde.

O primeiro parâmetro que o especialista pode definir é o número de conjuntos *fuzzy* que compõem cada partição e seus respectivos termos linguísticos associados. Partindo deste, são apresentadas algumas opções de particionamento automático para que o especialista possa escolher:

- **Uniforme:** Consiste em uma partição em que o UD da variável é dividido uniformemente entre o número de conjuntos determinado. A Figura 2.9 mostra uma partição uniforme com cinco conjuntos *fuzzy*;
- **K-means:** Consiste em uma partição induzida através do algoritmo de agrupamento K-means, onde os centroides dos K agrupamentos descobertos representam os pontos de máximo de conjuntos *fuzzy* triangulares.
- **Equifrequente:** Consiste em uma partição que tenta equilibrar a mesma quantidade de instâncias do conjunto de dados (em que há a máxima pertinência) para cada CF da partição.
- **Definida pelo especialista:** Consiste em uma partição em que o especialista define manualmente os pontos de máximo dos conjuntos *fuzzy*, bem como o UD da variável.

Independente do tipo de particionamento escolhido pelo especialista, é apresentado em tempo real, como um subsídio adicional ao processo de modelagem, o gráfico da partição, bem como a quantidade de instâncias da amostra de dados com máxima pertinência para cada conjunto definido.

5.2.1 Aprendizado dos parâmetros desconhecidos

Ao final do processo de definição de parâmetros por parte do especialista, podem existir parâmetros assinalados como *'desconhecido'* que necessitam ser definidos para que se tenha uma BD *fuzzy* completa. O desafio nesta tarefa é que, na maioria dos casos, o espaço de busca é muito grande, sendo inviável usar AGMO (que é a técnica escolhida para aprender a BR posteriormente) para avaliar a acurácia de cada combinação de parâmetros possível para as partições *fuzzy* das variáveis que compõem a BD.

Para contornar este problema, foi proposto um algoritmo (Algoritmo 5.1) com complexidade linear, que ao invés de utilizar AGMO, adota a técnica de aprendizado da base de regras proposta por Wang&Mendel (WANG; MENDEL, 1992; WANG, 2003), para identificar que tipo de partição ofereceria uma melhor acurácia ao SFBR.

O princípio do algoritmo é variar, para cada parâmetro desconhecido, as possíveis opções de quantidade de conjuntos na partição (caso não tenha sido definida pelo especialista) e tipo de partição (uniforme, k-mean e equifrequente), mantendo as definições que já foram feitas pelo especialista. Para que essa iteração não resulte em uma complexidade exponencial, cada opção é testada somente uma vez para o parâmetro desconhecido corrente, sendo o restante deles mantido fixo como do tipo uniforme com 3 conjuntos *fuzzy* (a opção mais simples possível), visando não influenciar o teste. Ao final, a BD resultante é definida pela opção que apresentar a melhor acurácia para cada parâmetro desconhecido.

Como mecanismo de validação, o conjunto de dados fornecido é dividido em duas partes. A cada iteração, 80% das instâncias são usadas para construir a BR usando Wang&Mendel e 20% são usadas para o cálculo da acurácia do SFBR.

O produto final desta etapa é o particionamento *fuzzy* de cada variável de interesse do problema e seus respectivos termos linguísticos.

Algoritmo 5.1: APRENDE PARTIÇÕES DESCONHECIDAS

Dados:

$ListaDef$: Lista de definições do especialista para cada partição da BD.

$particao$: Objeto que representa uma partição *fuzzy*, contendo o $tipo$ e a quantidade de conjuntos ($qtdConj$) de uma partição.

DB : Lista com as definições aprendidas pelo algoritmo para cada partição da BD.

```
1  início
2      DB ← ListaDef
3      para cada particao ∈ ListaDef faça
4          se particao.tipo = 'desconhecido' então
5              se particao.qtdConj = 'desconhecido' então
6                  intervaloQtdConj ← {3..7}
7              senão
8                  intervaloQtdConj ← particao.qtdConj
9              fim
10             minErro ← ∞
11             para cada qtdConj ∈ intervaloQtdConj faça
12                 para cada tipo ∈ {'uniforme','k-means','equifrequente'} faça
13                     erro ←
                       CONSTRUAWANGMENDEL(ListaDef, particao, qtdConj, tipo)
14                     se erro < minErro então
15                         DB(particao).qtdConj ← qtdConj
16                         DB(particao).tipo ← tipo
17                     fim
18                 fim
19             fim
20         fim
21     fim
22 fim
23 retorna DB
```

Função ConstruaWangMendel(ListaDef, particaoAtual, qtdConj, tipo)

Dados:

ListaDef : Lista de definições do especialista para cada partição da BD.

particaoAtual : Objeto contendo a partição sendo testada.

qtdConj : Quantidade de conjuntos para ser testada.

tipo : Tipo de partição para ser testado.

DBtmp : Lista com as definições temporárias para cada partição da BD.

RBtmp : Base de regras temporária.

SFBRtmp : Sistema *Fuzzy* Baseado em regras temporário.

particao : Objeto contendo o tipo e a quantidade de conjuntos (*qtdConj*) de uma partição.

```
1  início
2      DBtmp ← ListaDef
       // define a partição atual com os parâmetros para teste
3      DBtmp(particaoAtual).qtdConj ← qtdConj
4      DBtmp(particaoAtual).tipo ← tipo
       // define as demais partições como uniforme com 3 CF
5      para cada particao ∈ DBtmp faça
6          se particao ≠ particaoAtual então
7              particao.qtdConj ← 3
8              particao.tipo ← 'uniforme'
9          fim
10     fim
11     Constrói BRtmp usando Wang&Mendel e DBtmp sobre 80% das instâncias
12     Constrói SFBRtmp usando DBtmp e BRtmp
13     Avalia a acurácia do SFBRtmp sobre 20% das instâncias
14 fim
15 retorna erro do SFBRtmp
```

5.3 Definição da base de regras inicial

Durante o processo de construção de um SFBR, sobretudo em contextos complexos, podem existir inúmeras BR que são igualmente boas para modelar o problema, pois o número de regras diferentes e combinações possíveis entre regras pode ser extremamente elevado. Entretanto, somente algumas poucas delas podem ser compatíveis com a semântica que o especialista espera

para o modelo.

Devido ao método escolhido para o aprendizado da BR (AGMO) ser, em sua natureza, estocástico, ele pode convergir para um grupo de soluções que, apesar de boas do ponto de vista das métricas usadas na busca, não sejam as soluções compatíveis com o conhecimento do especialista que desejamos. Então é importante que o algoritmo seja, de alguma forma, direcionado para a busca de um resultado que tenha esta compatibilidade.

Uma maneira possível de efetuar este direcionamento é adicionar à população inicial do AGMO uma base de regras definidas pelo especialista e que caracterize esta compatibilidade. Isto se explica pois, dado que o processo de construção da população inicial cria indivíduos aleatoriamente, a tendência é que esta base de regras do especialista esteja entre os indivíduos mais aptos, influenciando as gerações subsequentes e, consequentemente, a convergência para soluções mais compatíveis.

O objetivo desta etapa, então, é a criação desta base de *Regras do Especialista* (RE). Nela, a base de dados construída na etapa anterior deve ser utilizada para construir este conjunto inicial de regras, que alimentará o processo de aprendizado genético. Este conjunto de regras também será armazenado para ser utilizado mais adiante no processo, durante a etapa de conciliação da base de regras.

Para esta construção, o especialista deve definir regras que expressem as tendências gerais do sistema e os casos particulares que ele julga caracterizarem o correto funcionamento do modelo. Nesta parte, o especialista não deve tentar enumerar todas as regras do modelo, muito menos todas as regras possíveis; a preocupação é encontrar regras que expressem o seu *"know-how"*.

Para auxiliar o especialista durante esta construção, será possível efetuar inferências neste SFBR sendo definido, bem como avaliar seu comportamento sobre o conjunto de dados fornecido. O produto final desta etapa é uma base de regras inicial composta pelas RE.

5.4 Aprendizado genético da base de regras

O objetivo desta etapa é realizar um processo de aprendizado da BR por meio de um AGMO, utilizando a BD já definida e uma população inicial contendo as RE.

Segundo o comparativo de performance feito em Ishibuchi, Nakashima e Nojima (2010), entre os AGMO NSGA-II (e dois variantes) e o MOEA/D, aplicados ao aprendizado de BR de SFBR, o NSGA-II foi o que apresentou a melhor performance. Já no processo de aprendizado

genético proposto em (CAMARGO, 2012), não houve diferença significativa de performance entre o algoritmo NSGA-II e o SPEA2. Estando estes três AGMO entre os mais utilizados para o aprendizado de SFBR (SHUKLA; TRIPATHI, 2012a, 2012b), optamos por adotar o NSGA-II, visto ser uma opção bastante consolida. Espera-se, também, que a heurística do elitismo (onde as melhores soluções sempre farão parte da população subsequente), adotada pelo NSGA-II, possa manter na população final soluções que incluam as RE, facilitando o processo de conciliação efetuado posteriormente.

5.4.1 Codificação dos Cromossomos

O primeiro passo ao se utilizar um AG na busca de soluções para um problema é determinar como uma solução será representada de maneira que permita as operações genéticas serem aplicadas pelo algoritmo, ou seja, como será a codificação dos cromossomos (ver seção 3.2).

Nesta abordagem, optamos por uma codificação dos cromossomos baseada na abordagem de Pittsburg (cada cromossomo representa uma base de regras inteira), pois ela permite avaliar o comportamento das regras em conjunto em uma BR, ao invés do funcionamento isolado de cada regra, como nas outras abordagens mais frequentemente utilizadas (Michigan e Interativa). Outra vantagem da abordagem de Pittsburg é que ela tem como resultado um conjunto de soluções, ao invés de uma única solução, o que é uma característica chave para incluir a preferência do especialista no processo de modelagem, pois permite que ele escolha a solução mais adequada segundo seu ponto de vista.

Cada gene do cromossomo utilizará valores inteiros que representarão o índice de cada conjunto *fuzzy* na partição *fuzzy* de uma variável, sendo o valor zero (‘0’) usado para a condição *don’t care*, onde a variável não é usada. Desta maneira, uma regra é codificada como uma sequência de inteiros de tamanho fixo n que representam os conjuntos fuzzy usados nos antecedentes e consequente da regra, onde n corresponde ao número de variáveis do SFBR. Uma BR será então representada pelo conjunto de todas as suas regras justapostas em sequência.

A quantidade máxima de regras que uma BR poderá possuir será definida pelo especialista como um parâmetro (K) para o algoritmo, logo, o cromossomo também terá um tamanho fixo N, definido como em (5.1):

$$N = n \cdot K \tag{5.1}$$

Neste modelo de codificação, a condição que representa a ausência de uma regra na BR

é dada quando a variável no consequente da regra é codificada com o valor zero (‘0’). Desta maneira, uma BR não precisa sempre ter o tamanho máximo de regras. Um exemplo ilustrativo da representação do cromossomo adotada pode ser visto na Figura 5.3, que codifica um sistema contendo 4 variáveis de entrada e uma variável de saída.

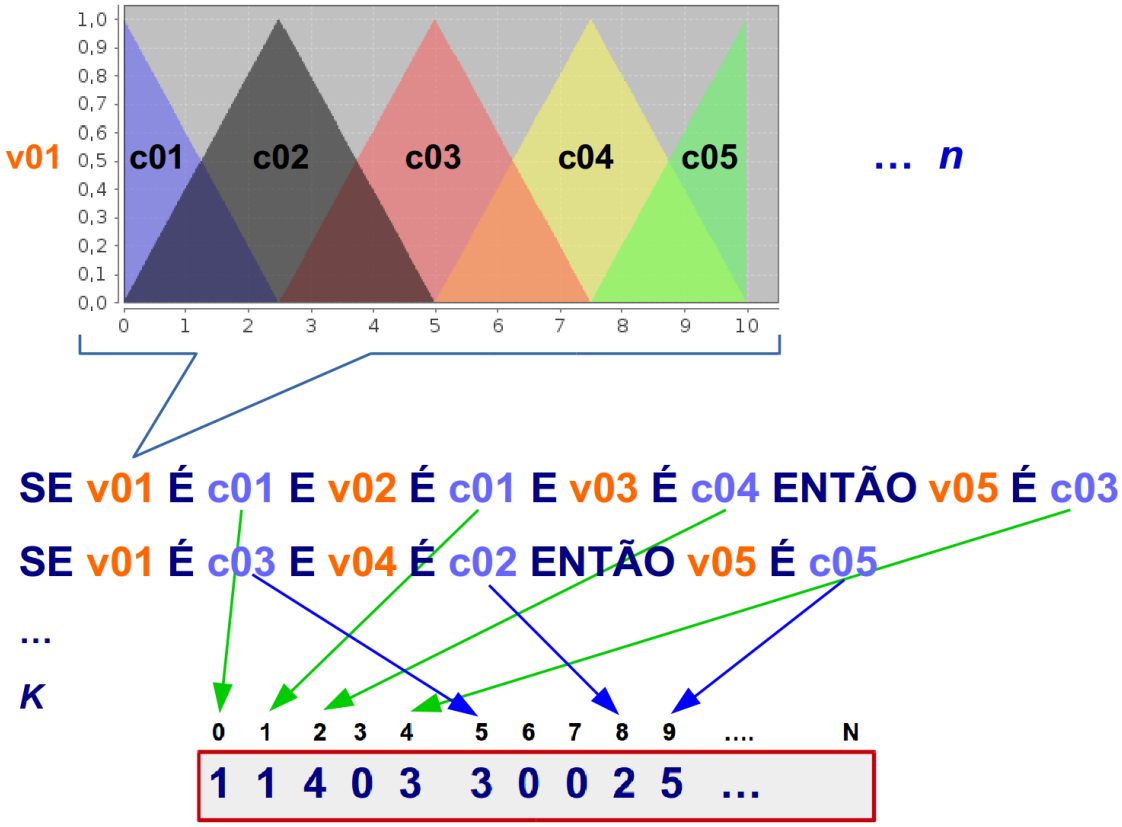


Figura 5.3: Exemplo da codificação do cromossomo adotada para representar uma BR no AGMO.

5.4.2 Funções Objetivo

Nossa intenção ao adotar um AGMO é possibilitar o balanceamento entre interpretabilidade e acurácia nas soluções (BR) encontradas. Desta maneira, estas serão as duas funções objetivo que adotaremos para o algoritmo, sendo responsáveis por avaliar cada cromossomo $C_i, i = 1..K$, como definidas a seguir:

- **Interpretabilidade da BR:** Alonso et al. (2011) cita como as principais métricas adotadas para medir a complexidade de uma BR o número de regras e o número de antecedentes das regras. Estes componentes serão contabilizados na BR que um indivíduo representa, para compor um índice de interpretabilidade, como definido em (5.2):

$$interpretabilidade(C_i) = \frac{1}{2}\left(\frac{numR}{K}\right) + \frac{1}{2}\left(\frac{\sum_{j=1}^{numR} Ant_j}{K \cdot E}\right) \tag{5.2}$$

em que *numR* corresponde ao número de regras que compõem a BR, K corresponde ao número máximo de regras na BR, E corresponde ao número de variáveis de entrada do SFBR e Ant_j corresponde ao número de antecedentes da regra j.

- **Acurácia da BR:** a métrica adotada irá depender da natureza do problema que está sendo modelado. Caso seja um problema de classificação, será adotada a taxa de classificação correta (TCC) do SFBR construído utilizando a BD Inicial e a BR codificada por C_i, computada sobre o conjunto de dados fornecido, como em (5.3):

$$acurácia(C_i) = TCC(C_i) = \frac{A}{M} \tag{5.3}$$

em que M é a quantidade total de instâncias existentes no conjunto de dados fornecido e A é a quantidade de instâncias onde o SFBR acertou a classificação.

Caso seja um problema de regressão, será adotada a raiz quadrada do erro quadrático médio (REQM) do SFBR, computada sobre o conjunto de dados fornecido, como em (5.4):

$$acurácia(C_i) = REQM(C_i) = \sqrt{\frac{1}{M}\sum_{j=1}^{M}(\hat{x_j} - x_j)^2} \tag{5.4}$$

em que M é o número de instâncias existentes no conjunto de dados fornecido, x_j é o valor predito pelo SFBR para a j-ésima instância e $\hat{x_j}$ é o valor da variável para a j-ésima instância no conjunto de dados fornecido.

5.4.3 População Inicial

Devido aos problemas ocasionados pelo vasto espaço de busca (ver seção 5.1) decorrente da tarefa de aprender a BR de um SFBR, qualquer iniciativa que tente restringir o tamanho deste espaço é muito interessante para esta abordagem.

Geralmente, a população inicial de um AG é gerada de maneira aleatória, onde os genes que formam o cromossomo são sorteados de um conjunto de genes possíveis. No modelo de codificação adotado nesta abordagem, cada gene representará um conjunto *fuzzy* de uma

variável, que justapostos, formarão as regras da BR. Sendo assim, uma maneira de gerar um cromossomo é gerar regras de maneira aleatória, respeitando a quantidade de conjuntos de cada variável neste sorteio, até que a quantidade de regras atinja o tamanho máximo da BR determinado pelo especialista.

Apesar da simplicidade do procedimento descrito, uma desvantagem que ele apresenta é que, dada a vasta combinação possível entre os conjuntos *fuzzy* das variáveis que podem compor uma regra aleatória, certamente serão formadas regras que quando avaliadas sobre o conjunto de dados fornecido não terão um grau de ativação relevante para nenhuma instância. Isto implica que somente um pequeno grupo de regras que podem ser geradas aleatoriamente de fato terão influência na convergência do algoritmo, sendo o restante delas desnecessárias para a busca.

Observado isto, buscamos restringir o espaço de busca por meio da utilização de um grupo de regras não aleatório para a geração dos cromossomos que irão compor a população inicial. Este grupo, que chamaremos *Regras do Conjunto de Dados* (RCD), será construído somente com regras que terão algum grau de ativação quando avaliadas sobre o conjunto de dados. De fato, armazenaremos esse conjunto de regras e o usaremos para restringir a busca não só na população inicial, mas no restante do AG também.

Para a geração deste conjunto de regras usaremos um algoritmo (Algoritmo 5.2) semelhante ao método de Wang&Mendel, porém, sem o seu último passo (de desambiguação), pois deixaremos que o próprio AGMO decida qual regra é mais adequada. De maneira simplificada, o que o algoritmo faz é criar uma regra para cada instância do conjunto de dados, onde cada variável (termo da regra) terá o valor do conjunto *fuzzy* que tiver o grau de pertinência máximo para o valor da variável na instância. Se uma regra já existir nas RCD, então ela não é novamente adicionada, de maneira que não existirão regras repetidas ao final.

De posse das RCD, os cromossomos da população inicial podem ser gerados por uma amostragem aleatória e sem reposição de regras das RCD, até que o *tamanho da população* (parâmetro definido pelo especialista) seja atingido. Além destes cromossomos, uma parte da população inicial (definida por um parâmetro) será composta por cromossomos que codificam a BR definida pelo especialista (RE). Esta população inicial mista trará o efeito que desejamos de manutenção da semântica definida pelo especialista, mas também abrirá espaço para uma variabilidade genética na população, o que é muito importante para evitar ótimos locais na convergência dos AG.

Algoritmo 5.2: CRIA REGRAS DO CONJUNTO DE DADOS

Dados:

Conjunto_de_Dados : É o conjunto de dados fornecido.

Variável : Objeto que representa uma variável do do *SFBR*, contendo o nome e o *tipo* da variável.

Regra : Objeto que representa uma regra do *SFBR*, contendo um *termo* linguístico para cada variável.

Instância : Objeto que representa uma instância do *Conjunto_de_Dados*, contendo o *valor* para cada variável.

CF : Objeto que representa um conjunto fuzzy de uma variável, contendo o seu *nome* e sendo capaz de calcular o grau de *pertinência* de um valor ao conjunto.

```
1  início
2      RCD ← novo conjunto de regras vazio
3      para cada Instância ∈ Conjunto_de_Dados faça
4          Regra ← nova regra vazia
5          para cada Variável ∈ SFBR faça
6              se Variável.tipo = 'uniforme' então
7                  Regra.termo(Variável) ← Instância.valor(Variável)
8              senão
9                  maxP ← 0
10                 para cada CF ∈ Variável faça
11                     se CF.pertinência(Instância.valor(variável)) > maxP então
12                         Regra.termo(Variável) ← CF.nome
13                     fim
14                 fim
15             fim
16         fim
17         se Regra ∉ RCD então
18             RCD ← RCD ∪ Regra
19         fim
20     fim
21 fim
22 retorna RCD
```

5.4.4 Operadores Genéticos

Buscando reduzir o espaço de busca do AGMO também durante as operações genéticas que modificam os cromossomos de sua população, foi necessária a customização dos operadores genéticos adotados no algoritmo, a fim de refletir as restrições impostas nas RCD. Adicionalmente, heurísticas foram criadas para permitir o aprendizado de regras mais simples, que não tenham necessariamente um termo para cada variável do SFBR nos antecedentes da regra (como nas RCD). Isto possibilita melhoria tanto na acurácia quanto na interpretabilidade das soluções encontradas. A seguir, descreveremos os operadores utilizados:

- **Seleção:** O operador de seleção padrão do NSGA-II foi mantido. Ele seleciona os indivíduos para a nova população por meio de torneio binário, usando como métrica o nível de cada solução na fronteira de Pareto, ou se elas fizerem parte da mesma fronteira, a com maior distância de multidão (DEB et al., 2002);

- **Mutação:** Foi criado um operador semelhante à mutação uniforme, porém, ao invés de um gene (CF) do cromossomo ser trocado de maneira aleatória, uma regra inteira é substituída no cromossomo (dada uma probabilidade definida pelo especialista) por uma regra formada a partir das RCD. O ponto onde ocorrerá a mutação é sorteado somente entre as posições que ficam entre regras na codificação do cromossomo.

  A regra que for sorteada das RCD para efetuar a mutação não será novamente escolhida para efetuar outra mutação, na mesma solução, em uma geração subsequente, até que todas as regras das RCD tenham sido sorteadas (sorteio sem reposição). Isso minimiza a possibilidade de inserção de regras redundantes nas BR codificadas nas soluções.

  A regra sorteada das RCD também sofrerá um processo de simplificação antes de ser inserida no cromossomo, onde, para cada termo do antecedente da regra, existirá a possibilidade (definida pelo especialista como um parâmetro) de o termo ser retirado da regra, ou seja, ser substituído por zero (0) na codificação da regra. A Figura 5.4 exemplifica o operador descrito.

- **Cruzamento:** Foi adaptado o cruzamento de 1 ponto, onde dada uma certa probabilidade (definida pelo especialista como um parâmetro), um ponto de cruzamento é escolhido aleatoriamente entre as posições que ficam entre regras na codificação de dois cromossomos pais, C_1 e C_2, e a porção compreendida depois deste ponto, em cada cromossomo, é trocada entre eles, para formar dois novos cromossomos filhos C_3 e C_4. A Figura 5.5 exemplifica o operador descrito.

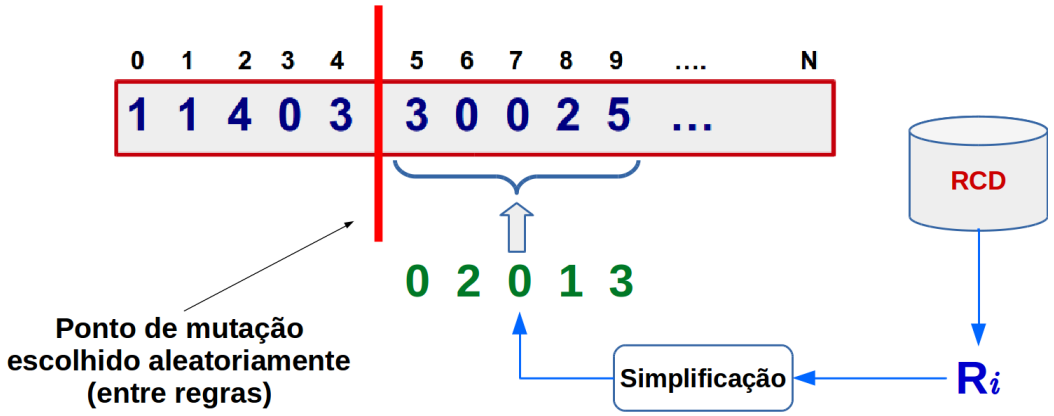


Figura 5.4: Exemplo do operador de mutação durante o aprendizado da BR, onde uma regra R_i sorteada das RCD, é simplificada e substitui a segunda regra de um cromossomo, em um sistema com 4 variáveis de entrada e uma de saída.

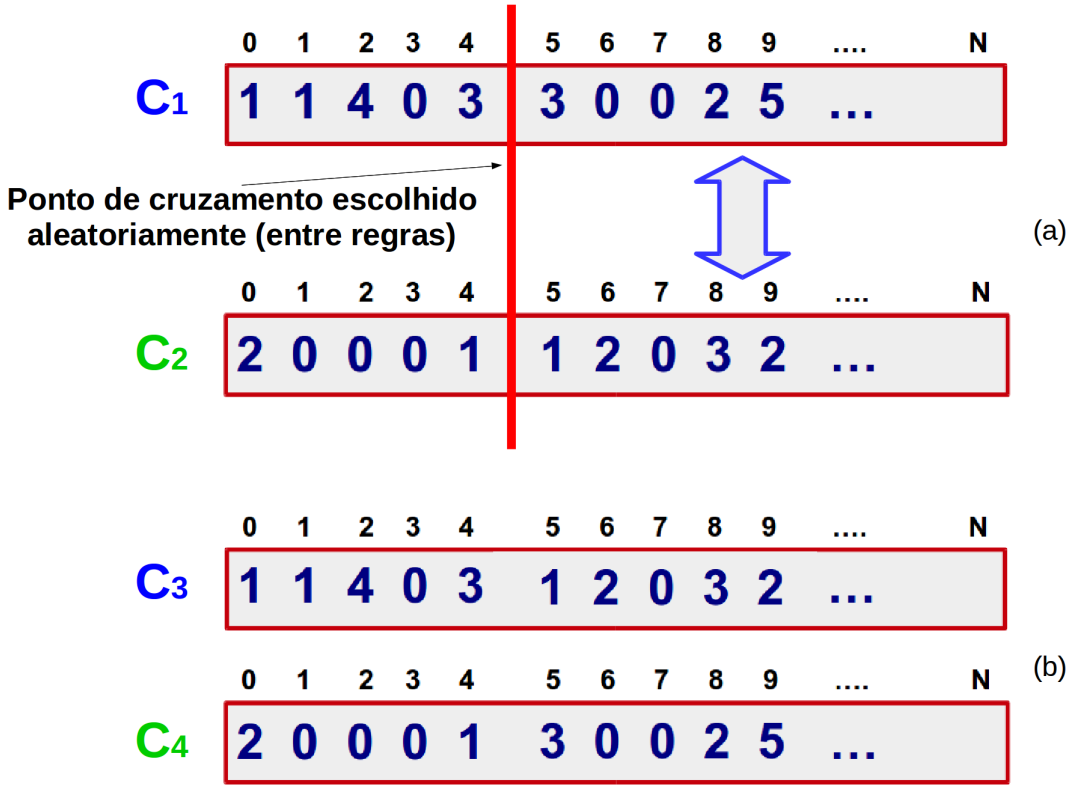


Figura 5.5: Exemplo do operador de cruzamento durante o aprendizado da BR, onde, em (a) um cromossomo pai C_1 sofre cruzamento com um cromossomo pai C_2 para formar dois novos cromossomos filhos, C_3 e C_4 (b), em um sistema com 4 variáveis de entrada e uma de saída.

- **Redução:** Foi introduzido um novo operador genético ao AGMO. O seu objetivo é possibilitar que, ao longo das gerações, as regras que compõem as BR codificadas pelos cromossomos sejam simplificadas ou removidas, viabilizando soluções mais interpretáveis e eficientes.

  O seu funcionamento é bastante simples: dada uma certa probabilidade (definida pelo especialista como um parâmetro), um ponto é escolhido aleatoriamente no cromossomo e o valor do gene é substituído por zero (0) na codificação, removendo o termo por ele representado na regra. Caso o ponto aleatório escolhido para substituição seja o consequente da regra, então a regra inteira é desconsiderada na BR codificada pelo cromossomo.

  Se, por acaso, o valor do gene no ponto escolhido já for igual a zero (0), então, será substituído o valor do ponto imediatamente subsequente na codificação, até que um valor diferente de zero seja encontrado. A Figura 5.6 exemplifica o operador descrito.

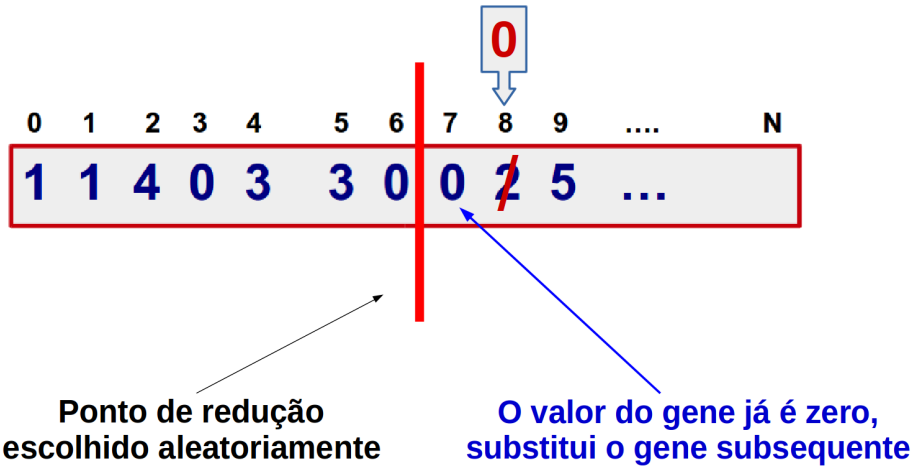


Figura 5.6: Exemplo do operador de redução, onde um termo é removido de uma regra codificada no cromossomo, em um sistema com 4 variáveis de entrada e uma de saída.

5.4.5 Simplificação da BR

Com o objetivo de maximizar a interpretabilidade das soluções (BR) resultantes do AGMO, após a execução deste, cada solução é submetida a um processo de simplificação realizado por um algoritmo (Algoritmo 5.3) de complexidade linear (proporcional ao tamanho do cromossomo), que toma como entrada a codificação da solução e produz uma solução simplificada de igual ou melhor acurácia.

Em linhas gerais, o funcionamento do algoritmo consiste em: aplicar, para cada gene da solução, o operador de redução descrito na seção 5.4.4, gerando uma solução temporária; construir um SFBR utilizando a BR codificada nesta solução e a BD definida pelo especialista; avaliar a

acurácia do SFBR construído e, caso a acurácia tenha diminuído, retornar a solução ao estado anterior.

Para otimizar sua execução, primeiramente o algoritmo tenta remover uma regra inteira da solução (removendo o consequente); caso a acurácia diminua, só então o algoritmo testará a remoção dos antecedentes desta regra.

Algoritmo 5.3: SIMPLIFICA SOLUÇÃO

Dados:

Sol : É a codificação da solução à ser simplificada.

Regra : Objeto que representa uma regra na codificação da solução, contendo os *antecedentes* e o *consequente*.

```
1  início
2      para cada Regra ∈ Sol faça
3          acurácia_anterior ← CONSTRÓIEAVALIASFBR(Sol)
4          SolTmp ← Sol
5          Sol ← APLICAOPERADORDEREDUÇÃO(Regra.consequente)
6          acurácia ← CONSTRÓIEAVALIASFBR(Sol)
7          se acurácia < acurácia_anterior então
8              Sol ← SolTmp
9              para cada antecedente ∈ Regra faça
10                 acurácia_anterior ← CONSTRÓIEAVALIASFBR(Sol)
11                 SolTmp ← Sol
12                 Sol ← APLICAOPERADORDEREDUÇÃO(Antecednte)
13                 acurácia ← CONSTRÓIEAVALIASFBR(Sol)
14                 se acurácia < acurácia_anterior então
15                     Sol ← SolTmp
16                 fim
17             fim
18         fim
19     fim
20 fim
21 retorna Sol
```

5.4.6 Validação e Apresentação dos Resultados

Qualquer processo de aprendizado que utilize um conjunto de dados para a construção de um modelo precisa estar aliado a alguma técnica que permita avaliar seu desempenho quando o modelo estiver em uma situação real de uso, ou seja, sendo alimentado por dados que não foram utilizados para o treinamento do modelo.

Uma das técnicas mais consolidadas para realizar tal validação é a chamada *"holdout"*, que consiste em separar o conjunto de dados em duas porções, uma que será utilizada para o treinamento do modelo e outra que será utilizada somente para avaliar seu desempenho. Entretanto, há duas desvantagens principais neste método, a primeira é que geralmente a captura de dados é uma tarefa cara e, algumas vezes, a quantidade de instâncias disponíveis é pequena, sendo ruim ainda ter que dividir o conjunto; a segunda desvantagem é que dependendo de como o conjunto de dados for dividido, pode ser que alguns exemplos (instâncias) que seriam muito importantes para o treinamento do modelo acabem ficando no conjunto de teste, prejudicando a obtenção de um bom modelo.

Uma alternativa para minimizar estes problemas é a validação cruzada (STONE, 1974). Nesta técnica, o conjunto de dados é dividido em N partes aproximadamente de mesmo tamanho e amostradas aleatoriamente, sendo efetuadas N construções do modelo, onde cada construção irá utilizar, de maneira alternada, uma parte para teste e o restante para treinamento, resultando que todas as instâncias acabarão sendo usadas para as duas tarefas em algum momento. O resultado da avaliação é dado então pela média do desempenho dos modelos das N partições.

Em problemas de classificação é possível minimizar ainda mais o risco de "exemplos importantes" serem distribuídos de maneira errônea entre as partições durante a divisão. Para isso, pode ser usado um processo de estratificação, onde cada partição receberá aproximadamente a mesma proporção de classes encontrada no conjunto de dados original.

Um número de partes (N) que é amplamente aceito como sendo o que oferece os melhores resultados é o número 10. Um motivo para tal é que um número maior que este deixaria menos de 10% das instâncias para teste. Neste caso, a técnica será então chamada de "validação cruzada estratificada de 10 partições" (*10-fold stratified cross-validation*) (WITTEN; FRANK; HALL, 2011).

O resultado da execução de um AGMO é um "conjunto" de soluções não-dominadas na fronteira de Pareto. Dito isto, fica evidente que a aplicação de validação cruzada neste contexto não é trivial, pois diferente de um algoritmo que retorna apenas um resultado, não é possível estabelecer um paralelo direto entre uma solução da fronteira construída em uma partição e

alguma outra solução construída em outra partição.

Uma solução adotada por muitos trabalhos, principalmente quando deseja-se comparar algoritmos, é escolher um ponto (solução) na fronteira resultante de cada partição e calcular a média do desempenho destas soluções como avaliação para o conjunto de resultados do algoritmo, mas não existe um consenso sobre a melhor maneira de se efetuar esta tarefa (ISHIBUCHI; NOJIMA, 2013a; ISHIBUCHI; MASUDA; NOJIMA, 2014).

Particularmente no contexto da abordagem aqui proposta, este método não é interessante, pois nosso maior interesse não é comparar o algoritmo, mas sim orientar o especialista na escolha de uma solução na fronteira que ofereça bom desempenho mas que mantenha as características desejadas por ele. Deste modo, avaliar o desempenho de somente um ponto na fronteira, daria subsídio ao especialista somente sobre este ponto.

A solução que adotamos para minimizar este problema consiste de validação cruzada (estratificada no caso de classificação) de N partições (onde N é definido pelo especialista), mas para o cálculo do desempenho médio (acurácia), são definidos algumas regiões e pontos de comparação ao longo de toda a fronteira, a saber:

- **Q1, Q2, Q3, Q4:** Correspondem à media de todas as soluções dos quartis da fronteira;
- **Primeira:** Corresponde à primeira solução, ou seja, a que tem menos erro dentre as soluções não dominadas da fronteira;
- **Mediana:** Corresponde à solução posicionada exatamente na metade da fronteira;
- **Mínima:** Corresponde à solução que, em termos absolutos, mais minimiza ambos os objetivos, ou seja, tem a menor distância euclidiana para a origem do plano cartesiano "Complexidade X Erro" (assinalada em azul na Figura 5.7);
- **Balanceada:** Corresponde à solução que, em relação às demais soluções da fronteira, mais minimiza ambos os objetivos, ou seja, tem a menor distância euclidiana para a origem do plano cartesiano formado pela solução com menor complexidade e pela solução com menor erro(assinalada em verde na Figura 5.7).

Adicionalmente às medidas de desempenho médio das regiões e pontos das soluções na fronteira, uma outra abordagem (adaptada de (ISHIBUCHI; NOJIMA, 2013b)) é também aplicada e apresentada ao especialista. O objetivo dela é mostrar o desempenho das soluções (BR) contendo um determinado número de regras, visto a regra ser o principal componente que adiciona complexidade à BR, sendo portanto, um fator decisório para o especialista.

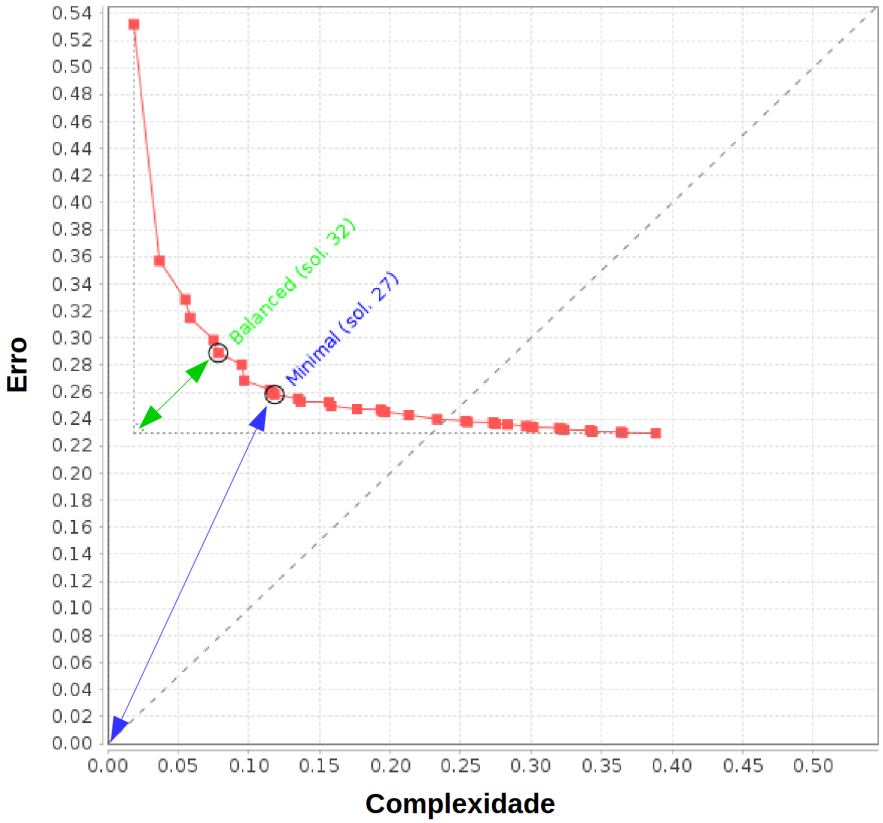


Figura 5.7: Exemplo de uma fronteira de Pareto resultante do AGMO. A determinação da solução Balanceada é assinalada em verde, enquanto a da solução Mínima em azul.

O procedimento consiste em calcular as médias de interpretabilidade e acurácia de todas as soluções com mesmo número de regras em cada partição e posteriormente calcular a média destas nas N partições. Outro dado útil apresentado é a quantidade de soluções com K regras encontradas nas N partições, visto que quanto maior esta quantidade, maior a possibilidade de esta ser uma solução mais robusta, pois foi encontrada mais vezes durante a construção das soluções.

Além das medidas de desempenho, o especialista será capaz de inspecionar os SFBR gerados a partir de cada solução apresentada na fronteira resultante do AGMO, bem como efetuar inferências e examinar os acertos e erros do modelo sobre o conjunto de dados fornecido.

De posse de todas os subsídios apresentados, a parte final desta etapa consiste na escolha, pelo especialista, de uma solução que melhor se adéque às suas preferências. Esta BR intermediária induzida dos dados será usada na etapa seguinte.

5.5 Conciliação da base de regras

Durante a etapa de aprendizado genético da BR, é possível que algumas das RE tenham desaparecido da BR resultante. Tal situação pode se apresentar em vários contextos: inicialmente, devido ao espaço de busca ser vasto e possibilitar inúmeras boas soluções (ver seção 5.3); outra possibilidade é o conjunto de dados fornecido não representar o problema de maneira fidedigna (ver Figura 5.1); pode ter havido a substituição de uma regra do especialista por uma ou mais regras equivalentes e que apresentavam melhor desempenho sobre o conjunto de dados; ou talvez o especialista não estava tão certo sobre alguma situação expressa em uma ou mais regras definidas nas RE; etc.

Com a finalidade de manter a semântica que o especialista deseja no modelo, aliada à acurácia que a BR aprendida por meio de aprendizado de máquina pode oferecer, faz-se necessário efetuar uma conciliação entre a BR resultante do AGMO e as RE, constituindo a BR Final. Este é o objetivo desta etapa do processo, que será composta por três passos, descritos a seguir:

- Primeiramente é feita uma análise onde cada RE é comparada com as regras da BR resultante do AGMO, identificando os conflitos existentes e armazenando-os em listas indexadas, uma para cada tipo de conflito;
- Cada conflito é apresentado ao especialista, para que este possa tomar uma ação de resolução do conflito. As ações possíveis irão depender do tipo do conflito, podendo ser: substituir a RE pela regra aprendida pelo AGMO; substituir a regra aprendida pelo AGMO

pela RE; ou manter as duas regras na BR Final. Devido os conflitos poderem ser alterados conforme alguma destas ações seja tomada, a cada passo, as listas indexadas de conflitos são atualizadas. A condição de parada é que não haja mais conflitos;

- A cada conflito resolvido pelo especialista, vai sendo formada a BR Final, que toma como base a BR aprendida pelo AGMO e a atualiza com as regras que o especialista decidiu manter.

Como subsídio para que o especialista possa tomar decisões mais embasadas sobre estes conflitos, serão apresentados em cada conflito: a mudança (acréscimo ou decréscimo) na acurácia resultante de cada possível opção à ser tomada; e o suporte que cada regra (do especialista e do AGMO) possui sobre o conjunto de dados fornecido. Estes indicadores podem guiar o especialista sobre a relevância das regras conflitantes, bem como sobre a compensação entre o ganho em acurácia ao se adicionar uma regra em relação à complexidade agregada a BR com esta adição. Também serão apresentadas a acurácia da BR formada pelas RE, da BR resultante do AGMO e da BR Final.

Os tipos de conflitos que podem acontecer no processo de conciliação, além da ausência das RE, podem ser conflitos lógicos estruturais decorrentes da junção de duas base de regras, como situações onde regras podem ser inconsistentes ou redundantes na BR. Na Tabela 5.2 são mostradas as situações em que cada conflito pode ocorrer entre duas regras na BR, considerando os antecedentes e consequentes de ambas.

Tabela 5.2: Conflitos que dever ser resolvidos durante a etapa de conciliação da BR (adaptada de (GUILLAUME; MAGDALENA, 2006)).

Antecedente	Consequente	
	$=$	$\neq$
$=$	Redundante	Inconsistente
$\subset$	Redundante	Inconsistente (especialização)
$\cap \neq \emptyset$	Parcialmente Redundante	Parcialmente Inconsistente
$\neq$	Sem problema	

Analisando cada situação da Tabela 5.2, uma ação é necessária para conciliar as regras:

- Caso os antecedentes e os consequentes das regras sejam iguais, as regras são redundantes e não é necessária nenhuma ação por parte do especialista, uma das regras é descartada;

- Caso os antecedentes sejam iguais e os consequentes diferentes, as regras são inconsistentes (no que tange interpretabilidade) e é mandatória uma escolha do especialista sobre qual regra manter;

- Caso o antecedente de uma regra esteja contido no antecedente de outra regra e os consequentes sejam iguais, as regras são redundantes e duas situações se apresentam:
  - Caso a regra mais geral seja a gerada pelos dados, não é necessária nenhuma ação por parte do especialista, a regra mais específica é descartada;
  - Caso a regra mais geral seja a do especialista, é apresentada uma escolha ao especialista sobre qual regra manter;
- Caso o antecedente de uma regra esteja contido no antecedente de outra regra e os consequentes sejam diferentes, não se trata de um problema de inconsistência em si, mas de uma especialização de uma regra para um contexto mais específico. Tal situação pode ser desejada pelo usuário, mas pode levar também a uma base de dados menos interpretável. É apresentada uma escolha ao especialista sobre quais regras manter, sendo possível manter as duas regras;
- As situações onde as regras são parcialmente redundantes ou parcialmente inconsistentes, ou onde os antecedentes são diferentes, não constituem um problema para a junção das regras e nenhuma ação é necessária por parte do especialista.

O produto desta etapa, é um SFBR que inclui as preferências do especialista e balanceia interpretabilidade e acurácia em uma BR gerada automaticamente.

5.6 Otimização genética das partições *fuzzy*

Esta é a última etapa do processo e, também, é opcional. Seu objetivo é tentar otimizar a BD quanto a sua acurácia, mantendo algumas restrições nas partições para que continuem interpretáveis. A BR não será alterada por este processo e no caso de problemas de regressão, a variável de saída também será otimizada.

Para possibilitar a flexibilidade necessária para tal otimização da BD, as PFF utilizadas na etapa de definição da BD são otimizadas por um processo baseado em AGMO que gera como resultado partições *fuzzy* transparentes. Será utilizado o algoritmo NSGA-II, adaptando a abordagem seguida em (CAMARGO, 2012).

5.6.1 Funções Objetivo

Nas funções objetivo do AGMO, assim como definimos na etapa de aprendizado genético da BR, nossa intenção é balancear a acurácia e a interpretabilidade nas soluções encontradas.

No caso da acurácia, as métricas utilizadas são idênticas às adotadas para a BR (ver seção 5.4.2).

Para a função objetivo que avalia a interpretabilidade, partiremos do princípio de que a partição mais interpretável sempre será a que o especialista já definiu. Será utilizado como métrica o índice de interpretabilidade semântica GM3M proposto em (GACTO; ALCALÁ; HERRERA, 2010), que é baseado na diferença entre o conjunto otimizado e o conjunto original.

O primeiro conceito que deve ser entendido para a adoção do GM3M é que, para manter as restrições das partições *fuzzy* transparentes (ver seção 4.3.2), o intervalo de variação permitida em relação aos conjuntos *fuzzy* originais durante o processo de otimização genética tem de respeitar alguns limites.

Para cada $CF_j = (a_j, b_j, c_j)$ na BD, onde (a,b,c) são os parâmetros de sua função de pertinência, $j = (1,...,m)$, e m é o número de CF existentes na BR, os intervalos de variação (ver Figura 5.8) são calculados como em (5.5):

$$\begin{aligned}
[I^l_{a_j}, I^r_{a_j}] &= \left[a_j - \tfrac{(b_j - a_j)}{2}, a_j + \tfrac{(b_j - a_j)}{2}\right] \\
[I^l_{b_j}, I^r_{b_j}] &= \left[b_j - \tfrac{(b_j - a_j)}{2}, b_j + \tfrac{(c_j - b_j)}{2}\right] \\
[I^l_{c_j}, I^r_{c_j}] &= \left[c_j - \tfrac{(c_j - b_j)}{2}, c_j + \tfrac{(c_j - b_j)}{2}\right]
\end{aligned} \tag{5.5}$$

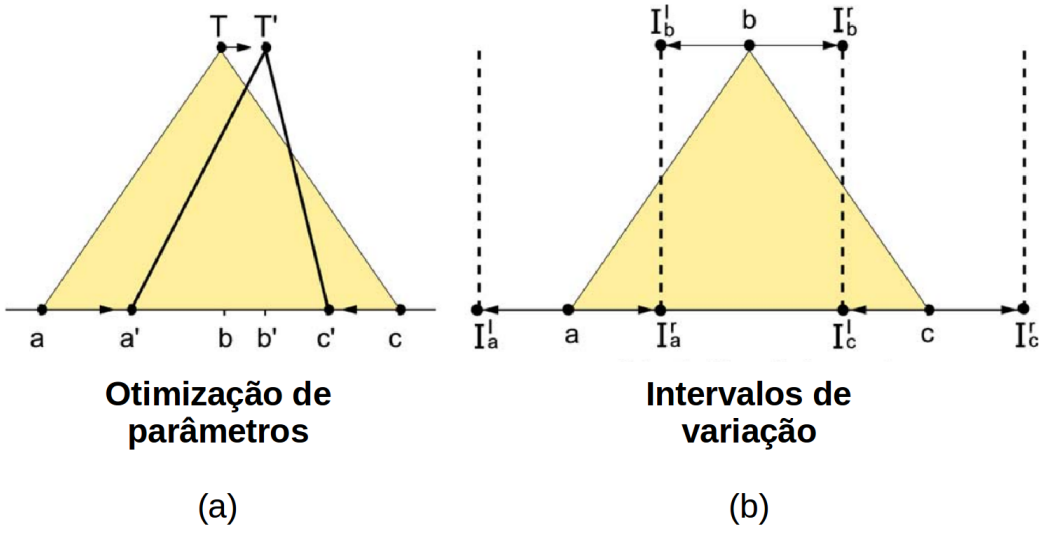


Figura 5.8: Intervalos de variação permitido pelo GM3M para a otimização de CF, onde (a) ilustra os parâmetros de um CF modificado e (b) ilustra os intervalos de variação permitidos(GACTO; ALCALÁ; HERRERA, 2010).

Como seu próprio nome (em inglês) já denota, *"Geometric Mean of Three Metrics"*, o GM3M utiliza três métricas, são elas:

- **Deslocamento do ponto central (δ):** Esta métrica controla o deslocamento do ponto central, baseando-se na distância normalizada entre o CF otimizado e o CF original, sendo calculada pelo deslocamento máximo observado na BD. Para cada CF_j na BD, o deslocamento é definido como em (5.6):

$$\delta_j = \frac{|b_j - b'_j|}{I} \tag{5.6}$$

em que I representa a variação máxima do ponto central do CF, definida como em (5.7):

$$I = \frac{I^r_{b_j} - I^l_{b_j}}{2} \tag{5.7}$$

Sendo o valor do deslocamento máximo δ^* é definido como em (5.8):

$$\delta^* = \max_j\{\delta_j\} \tag{5.8}$$

Como δ^* pode variar no intervalo entre zero (0) e um (1), onde 1 representa o maior deslocamento possível, é necessária uma transformação para que se possa usar a métrica como um subobjetivo do AGMO, definindo-a como em (5.9):

$$Maximizar\ \delta = 1 - \delta^* \tag{5.9}$$

- **Taxa de amplitude lateral (γ):** Esta métrica controla a forma do CF, baseando-se na relação entre os parâmetros esquerdo e direito do CF otimizado e o CF original. Sejam $esq\ S_j$ e $dir\ S_j$ as amplitudes da parte esquerda e direita do CF original e, respectivamente, $esq\ S'_j$ e $dir\ S'_j$ as amplitudes da parte esquerda e direita do CF otimizado, definidas como em (5.10):

$$\begin{aligned} esq\ S_j &= |a_j - b_j| \\ dir\ S_j &= |b_j - c_j| \\ esq\ S'_j &= |a'_j - b'_j| \\ dir\ S'_j &= |b'_j - c'_j| \end{aligned} \tag{5.10}$$

Sendo a taxa de amplitude lateral γ_j definida para cada CF_j da BD como em (5.11):

$$\gamma_j = \frac{\min\left\{\frac{esq\ S_j}{dir\ S_j}, \frac{esq\ S'_j}{dir\ S'_j}\right\}}{\max\left\{\frac{esq\ S_j}{dir\ S_j}, \frac{esq\ S'_j}{dir\ S'_j}\right\}} \tag{5.11}$$

Como γ_j pode variar no intervalo entre zero (0) e um (1), onde 1 representa que o formato dos CF originais foi preservado, a métrica γ pode ser utilizada como um subobjetivo do AGMO, definindo-a como em (5.12):

$$Maximizar\ \gamma = \min_j\{\gamma_j\} \tag{5.12}$$

- **Similaridade de área (ρ):** Esta métrica controla a área dos CF, baseando-se na relação entre a área do CF otimizado e o CF original. Sejam A_j a área do triângulo que representa o CF original e A'_j a área do triângulo que representa o CF otimizado, a similaridade de área ρ_j é definida como em (5.13):

$$\rho_j = \frac{\min\left\{A_j, A'_j\right\}}{\max\left\{A_j, A'_j\right\}} \tag{5.13}$$

Como ρ_j pode variar no intervalo entre zero (0) e um (1), onde 1 representa que a área dos CF originais foi preservada, a métrica ρ pode ser utilizada como um subobjetivo do AGMO, definindo-a como em (5.14):

$$Maximizar\ \rho = \min_j\{\rho_j\} \tag{5.14}$$

Definidas todas as métricas utilizadas, o cálculo do GM3M é dado por uma média geométrica, conforme definido em (5.15):

$$Maximizar\ \text{GM3M} = \sqrt[3]{\delta\gamma\rho} \tag{5.15}$$

O uso do mínimo e do máximo para computar as métricas do GM3M, assegura um nível mínimo de interpretabilidade em todas as funções de pertinência, visando sempre identificar o "pior caso". Isso evita convergência do AGMO para BD estranhas, onde quase todos os CF sofreram pouca distorção, mas alguns poucos foram significativamente alterados. Se esta heurística não fosse adotada, como a avaliação é uma média de todos os CF da BD, esta situação poderia passar despercebida.

5.6.2 Codificação dos Cromossomos

A codificação dos cromossomos seguirá a proposta de (GACTO; ALCALÁ; HERRERA, 2008). Possuirão tamanho fixo e manterão a mesma quantidade de conjuntos das partições já definidas. Os genes serão representados por números reais que representarão os parâmetros de uma função de pertinência triangular (ou a porção triangular de uma função de pertinência trapezoidal utilizada nos extremos do domínio da variável). Cada cromossomo representará uma base de dados inteira, que será a justaposição dos três parâmetro de cada CF, para cada variável do modelo também justapostas. A Figura 5.9 exemplifica a codificação adotada.

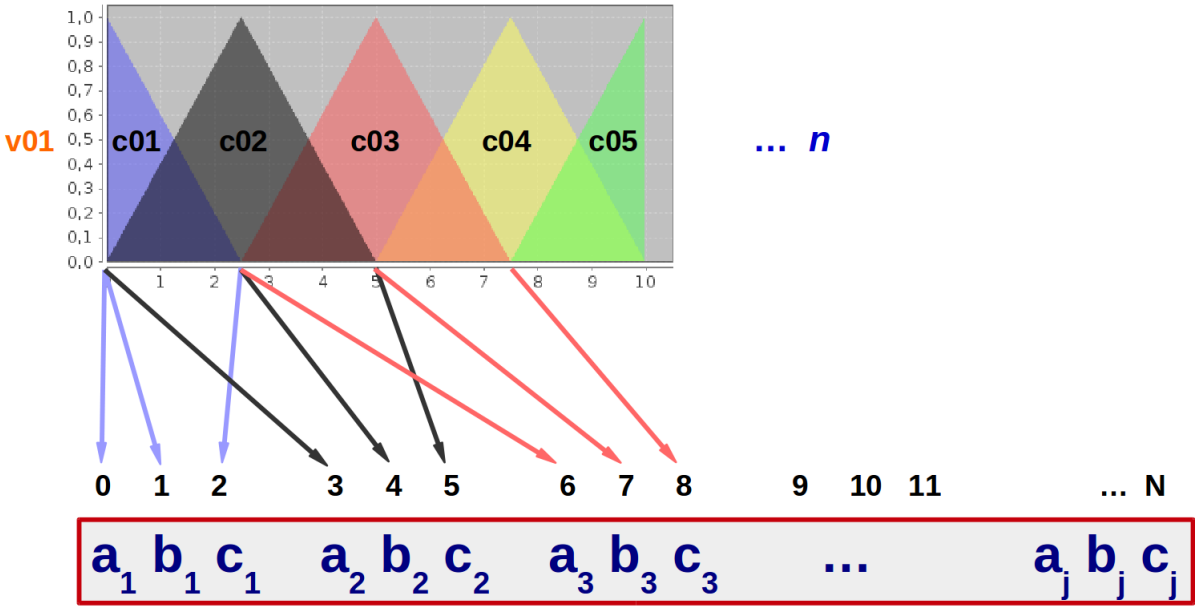


Figura 5.9: Exemplo da codificação do cromossomo adotada para representar uma BD no AGMO.

5.6.3 População Inicial

O primeiro cromossomo da população inicial será a codificação da BD definida pelo especialista, os demais cromossomos serão gerados aleatoriamente, respeitando em cada gene as restrições dos intervalos de variação dos CF definidos pelo GM3M em (5.5).

5.6.4 Operadores Genéticos

- **Seleção:** O operador de seleção padrão do NSGA-II foi mantido;
- **Mutação:** Foi adotada à mutação uniforme, onde dada uma probabilidade definida pelo especialista, o valor de um gene (parâmetro de um CF) é substituído por um valor ale-

atório no intervalo $I^l...I^r$. O ponto de mutação também é definido aleatoriamente na codificação do cromossomo. A Figura 5.10 exemplifica o operador descrito.

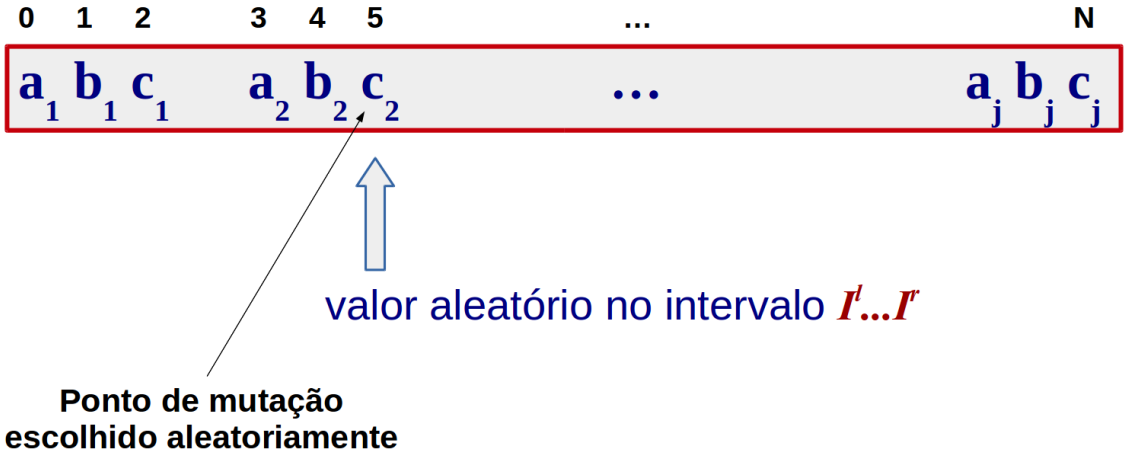


Figura 5.10: Exemplo do operador de mutação de um cromossomo durante a otimização da BD.

- **Cruzamento:** Foi adotado o cruzamento de 1 ponto, onde dada uma certa probabilidade (definida pelo especialista como um parâmetro), um ponto de cruzamento é escolhido aleatoriamente na codificação de dois cromossomos pais, C_1 e C_2, e a porção compreendida depois deste ponto em cada cromossomo, é trocada entre eles, para formar dois novos cromossomos filhos. A Figura 5.11 exemplifica o operador descrito.

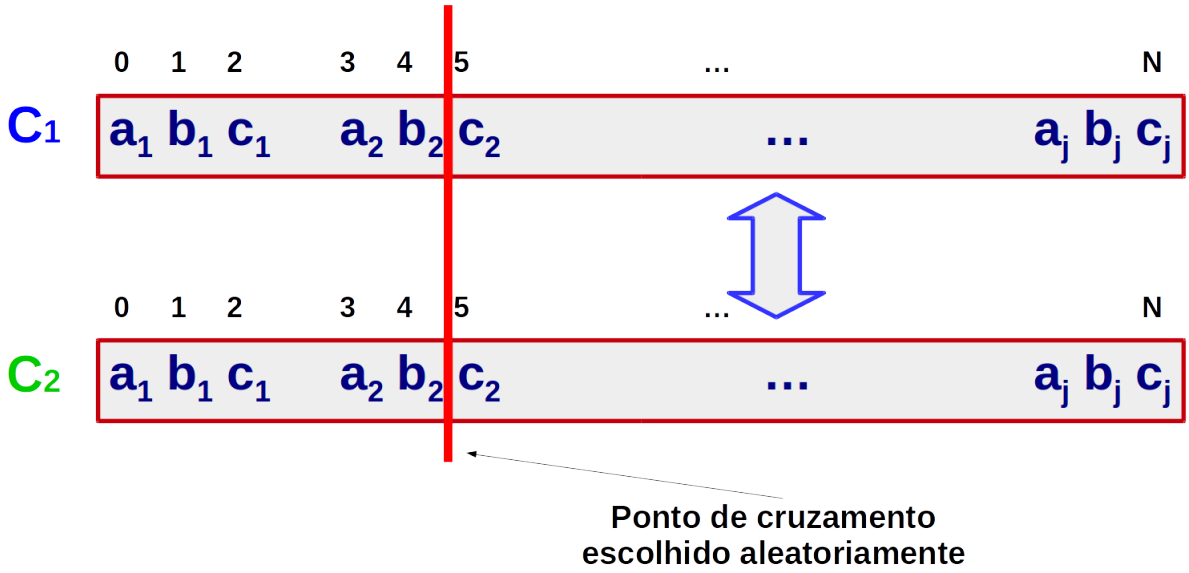


Figura 5.11: Exemplo do operador de cruzamento durante a otimização da BD, onde um cromossomo pai C_1 sofre cruzamento com um cromossomo pai C_2 para formar dois novos cromossomos filhos.

5.6.5 Validação e Apresentação dos Resultados

Para a validação e apresentação dos resultados do AGMO nesta etapa, foi adotada a mesma solução descrita na seção 5.4.6, consistindo de validação cruzada (estratificada no caso de classificação) de N partições e cálculo do desempenho médio (acurácia) para soluções nas regiões Q1, Q2, Q3 e Q4, bem como para as soluções Primeira, Mediana, Balanceada e Mínima.

De maneira semelhante à etapa de aprendizado da BR, o especialista será capaz de inspecionar os SFBR gerados a partir de cada solução apresentada na fronteira resultante do AGMO, bem como efetuar inferências e examinar os acertos e erros do modelo sobre o conjunto de dados fornecido.

A parte final desta etapa consiste na escolha pelo especialista, de uma solução que melhor se adéque às suas preferências, ou se nenhuma lhe agradar, é lhe dada a opção de recusar o processo e ficar com a BD definida anteriormente. O produto desta etapa é um SFBR resultante de todo o processo proposto.

Capítulo 6

A FERRAMENTA *EvoFuzzy*

Observada a complexidade da abordagem proposta (Capítulo 5), seria bastante trabalhoso adotá-la sem a utilização de uma ferramenta de *software* que oferecesse suporte para a aplicação de cada passo do processo. Além disso, muitas das tarefas realizadas em cada etapa são comuns à construção de qualquer SFBR utilizando dados, logo, mesmo que o objetivo não seja integrar o conhecimento do especialista na modelagem, as soluções propostas na abordagem podem ser utilizadas separadamente, de maneira que uma biblioteca de *software* (API[1]) que implemente tais tarefas seria bastante útil.

Diante disso, como complementação à abordagem, foi implementada uma ferramenta de *software* denominada **EvoFuzzy** que tem como objetivo viabilizar as soluções propostas e oferecer subsídio à correta aplicação do processo. Nas seções seguintes apresentaremos o detalhamento desta ferramenta.

6.1 Arquitetura

Buscando maximizar os contextos de utilização, algumas premissas foram estabelecidas durante a implementação da ferramenta. A primeira é quanto ao idioma das interfaces e código fonte, onde optamos pelo inglês, visando maximizar o potencial público alvo de utilização (portanto, todas as capturas de tela apresentadas neste capítulo são neste idioma). Com o mesmo objetivo, a linguagem de programação utilizada foi a linguagem Java, pois esta possibilita a execução em vários sistemas operacionais diferentes de maneira transparente.

Optou-se, também, por um modelo de arquitetura em camadas, onde a interface com o

[1]Do inglês *Application Program Interface* ou Interface de Programação de Aplicativos, é um conjunto de rotinas e padrões estabelecidos por um software para a utilização das suas funcionalidades por aplicativos que não pretendem envolver-se em detalhes da implementação do software, mas apenas usar seus serviços.

usuário é completamente desacoplada da lógica da aplicação, possibilitando a construção de uma API, uma interface de usuário via linha de comando e também uma interface gráfica para a ferramenta.

Com o propósito de garantir robustez ao *software*, algumas APIs bastante consolidadas e testadas foram utilizadas durante a implementação. Parte dos algoritmos de aprendizado de máquina utilizados nas etapas de seleção de atributos e definição das partições *fuzzy* são oriundas da API Weka (HALL et al., 2009), enquanto o mecanismo de inferência *fuzzy* adotado é proveniente da API jFuzzyLogic (CINGOLANI; ALCALA-FDEZ, 2012; CINGOLANI; ALCALÁ-FDEZ, 2013).

O AGMO utilizado nas etapas de aprendizado genético da BR e de otimização genética da BD foi uma customização da implementação paralelizada do NSGA-II apresentada em Nebro et al. (2008) e disponível na API JMetal (DURILLO; NEBRO; ALBA, 2010). Além do AGMO, o processo de validação cruzada também foi paralelizado, de maneira que em todas as etapas onde há construção de modelos, as *N* partições de validação são processadas em paralelo, conforme o número de núcleos disponíveis. Com isso pode-se obter uma diminuição considerável do tempo de processamento demandado, sobretudo pois a grande maioria dos computadores atuais possuem múltiplos núcleos.

6.2 API

Foi implementada uma API em Java com licença de software livre, que permite a incorporação de qualquer etapa da abordagem (bem como elementos da interface gráfica e linha de comandos) em aplicações de terceiros. Para sua utilização (assim como o restante da ferramenta), uma máquina virtual Java SE 7 ou superior é necessária.

A API está organizada em pacotes, um pacote contendo a lógica para cada etapa proposta, um dedicado à interface gráfica (*evofuzzy.gui*) e um (pacote raiz) contendo a lógica comum à todo o software. A estrutura completa é apresentada na Figura 6.1.

Para sua utilização, é necessário: adicionar o arquivo *evofuzzy.jar* ao CLASS_PATH da aplicação Java; definir os parâmetros de execução por meio das variáveis na classe *Config.class* e; executar a chamada do método à ser utilizado. Uma documentação no formato *Javadoc* descrevendo o pacote e as assinaturas dos métodos implementados é fornecida junto com a API.

O formato de arquivo adotado para importação de conjuntos de dados para a API (e demais

Figura 6.1: Estrutura de pacotes da API EvoFuzzy.

interfaces) é o de 'valores separados por vírgula' (CSV, do inglês *comma-separated values*) onde, os valores do atributo meta devem ser os últimos em cada linha de dados e valores numéricos devem utilizar o ponto (.) como separador decimal (padrão imperial de medidas).

O formato de arquivo adotado para importação e exportação dos SFBR gerados pela API (e demais interfaces) é o FCL (*Fuzzy Control Language*) definido pela norma *IEC 61131 parte 7* [2] como um padrão para interoperabilidade de sistemas *fuzzy*. É possível incorporar SFBR definidos em arquivos FCL à aplicações Java usando a própria API da EvoFuzzy. Adicionalmente, é possível exportar os modelos gerados como código fonte C++.

6.3 Interface de linha de comando

Apesar de este tipo de interface não ser a preferida da grande maioria das pessoas, ela é necessária em alguns contextos, como por exemplo quando se deseja integrar a ferramenta à outras via *scripts* em um processo, ou quando deseja-se executar a ferramenta em uma máquina que não oferece interface gráfica (como uma máquina com sistema operacional UNIX, Linux e seus variantes).

Particularmente devido ao alto custo computacional dos AGMO, a interface de linha de comando é desejável para possibilitar a utilização da ferramenta em máquinas com maior poder de processamento, como servidores e *clusters*, que geralmente não oferecem interface gráfica.

Para utilização da interface de linha de comando, o usuário deve definir os parâmetros de execução no arquivo de definições *config.properties* e então passar o parâmetro *-cmd* na chamada da aplicação, como a seguir:

[2] IEC (*International Electrotechnical Commission*) é uma organização internacional responsável pela criação de padrões para a indústria eletro-eletrônica.

```
> java -jar evofuzzy.jar -cmd
```

6.4 Interface Gráfica

Constitui a interface preferencial da ferramenta, tendo sido implementada como uma aplicação *desktop* utilizando a tecnologia Java Swing. Seu ambiente de trabalho é multitarefa, ou seja, os processos são executados em segundo plano, enquanto a interface continua disponível ao usuário. Uma barra de progresso e uma janela de *log* mostram o andamento e os detalhes do processo sendo executado. Os elementos gráficos da interface são redimensionados dinamicamente para acompanhar o tamanho da janela da aplicação; eles também oferecem ajuda contextual (dicas) quando o ponteiro do mouse fica estacionado sobre eles. Além disso, cada tela da aplicação oferece um texto de ajuda explicando sua funcionalidade.

A interação com o usuário é organizada por meio de abas, cada uma representando uma das etapas da abordagem proposta. Cada etapa pode ser executada de maneira sequencial ou independente, oferecendo interface de importação de arquivos contendo as entradas e exportação de arquivos contendo as saídas. A seguir descreveremos o funcionamento de cada uma destas abas na interface da ferramenta. Utilizaremos como exemplo de entrada o conjunto de dados *Wine* disponível na UCI.

A primeira aba (*Attribute Selection*) corresponde à etapa de seleção de atributos (ver seção 5.1), ilustrada na Figura 6.2. Aqui o usuário deve carregar um arquivo contendo o conjunto de dados à ser analisado e clicar no botão *Analyze* para iniciar o processo de análise por meio dos algoritmos utilizados com este fim (ver seção 5.1.1). Um maior detalhamento do processo executado pode ser visualizado clicando-se na barra de progresso, o que mostrará a janela de *log* da ferramenta (Figura 6.3). Será então apresentada uma tabela com os resultados do processamento, onde o usuário (especialista ou com auxílio deste) deve escolher quais atributos devem ser mantidos clicando na caixa *Keep* posicionada na última coluna da tabela. Feito isto, ele pode salvar um conjunto de dados com os atributos selecionados para uso posterior ou seguir para a próxima etapa clicando no botão *'Use in next step'*.

A aba (*Partitioning*) corresponde à etapa de definição das partições *fuzzy* (ver seção 5.2). Aqui o usuário deve definir as partições *fuzzy* das variáveis do SFBR. É mostrado no painel *Variables* (Figura 6.4) uma tabela contendo as variáveis à serem definidas. Selecionando uma delas, na sub-aba *Variable Summary*, são apresentados um sumário estatístico da variável, um diagrama de caixa e um histograma (onde as cores representam a proporção das classes). Na sub-aba *Variable Partition* (Figura 6.5) o usuário pode definir os parâmetros do particionamento

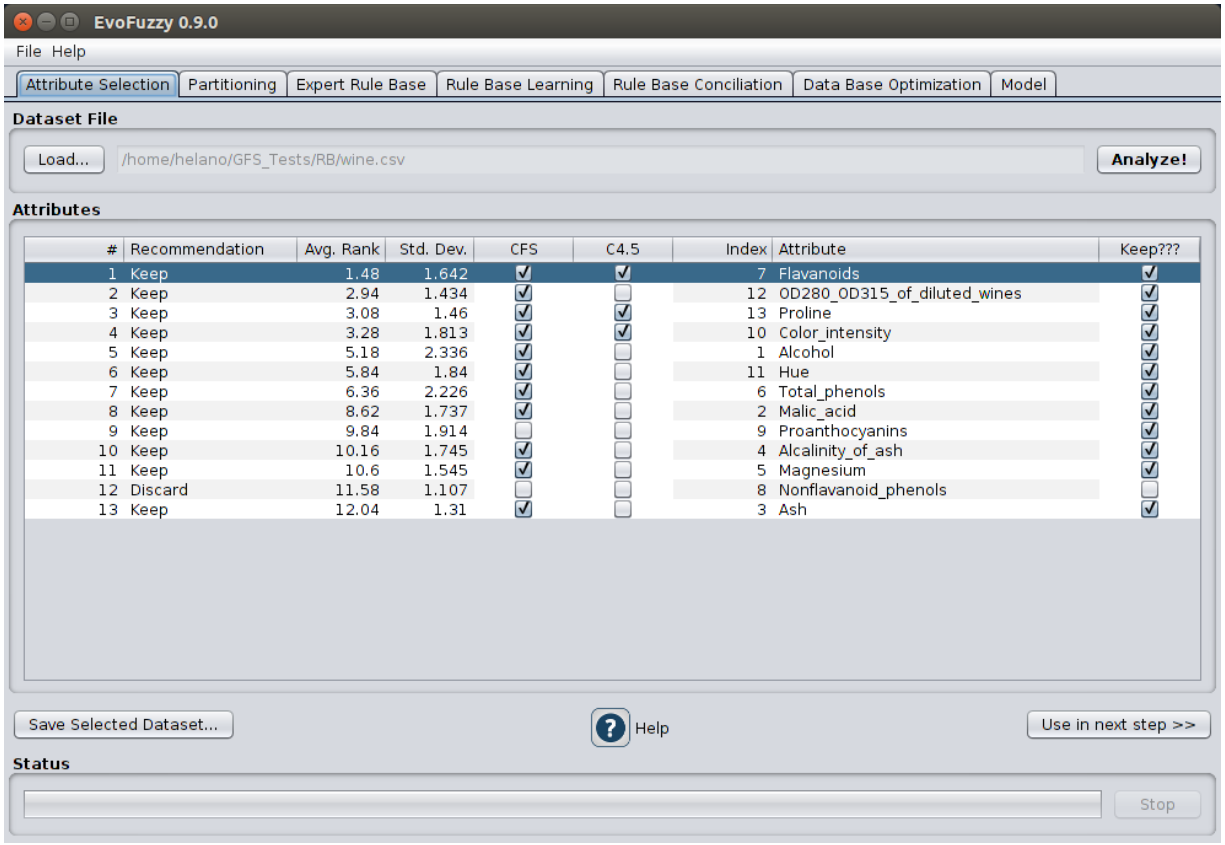


Figura 6.2: Aba da etapa de seleção de atributos, mostrando o resultado do processamento.

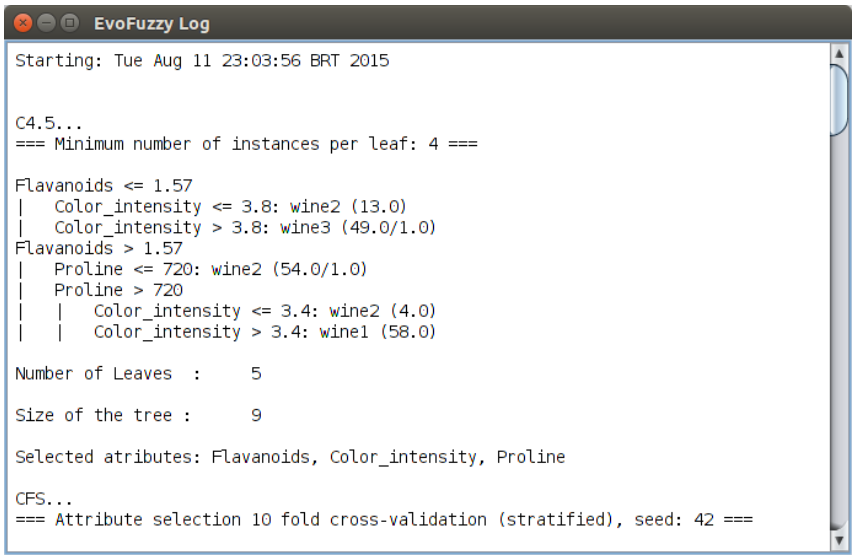


Figura 6.3: Janela de log mostrando detalhamento do processo de seleção de atributos.

fuzzy da variável selecionada manualmente ou automaticamente. Ao final do processo, caso o usuário (especialista) não consiga definir as partições de algumas variáveis, pode-se aprender as partições por meio de aprendizado de máquina (ver seção 5.2.1) clicando no botão *Learn Unknown Partitions*. Com todas as definições concluídas, pode-se salvar um arquivo de modelo das partições para uso posterior ou seguir para a próxima etapa clicando no botão *'Use in next step'*.

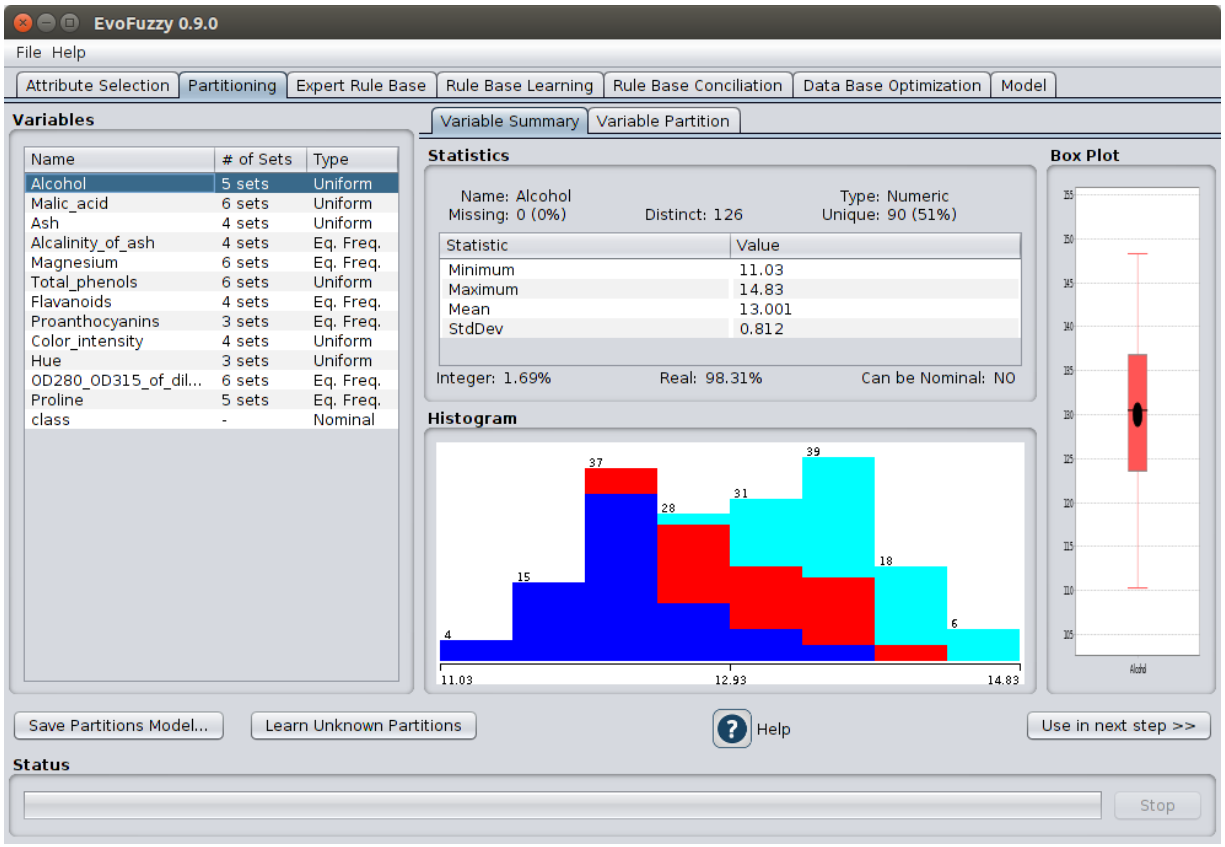


Figura 6.4: Aba da etapa de definição das partições *fuzzy*, mostrando a sub-aba de sumário da variável.

A aba (*Expert Rule Base*) corresponde à etapa de definição da BR Inicial (ver seção 5.3). Aqui o usuário (especialista) deve definir as RE que serão utilizadas mais adiante no processo. É mostrado no painel *Variables* (Figura 6.6) uma tabela contendo as variáveis do SFBR. Selecionando uma delas, no painel *Sets*, são apresentados os conjuntos *fuzzy* da respectiva variável. A construção de uma regra se dá ao escolher este dois parâmetros e clicar no botão *'add this term'*, até que sejam definidos os antecedentes e consequente desejados, quando então pode-se adicionar a regra à BR clicando no botão *'Add to Rule Base'*. A qualquer momento o usuário pode examinar o SFBR sendo construído clicando no botão *'Examine Expert Model'*, o que o levará à aba *Model*. Com as RE definidas, pode-se salvar um arquivo de modelo do especialista para uso posterior ou seguir para a próxima etapa clicando no botão *'Use in next step'*.

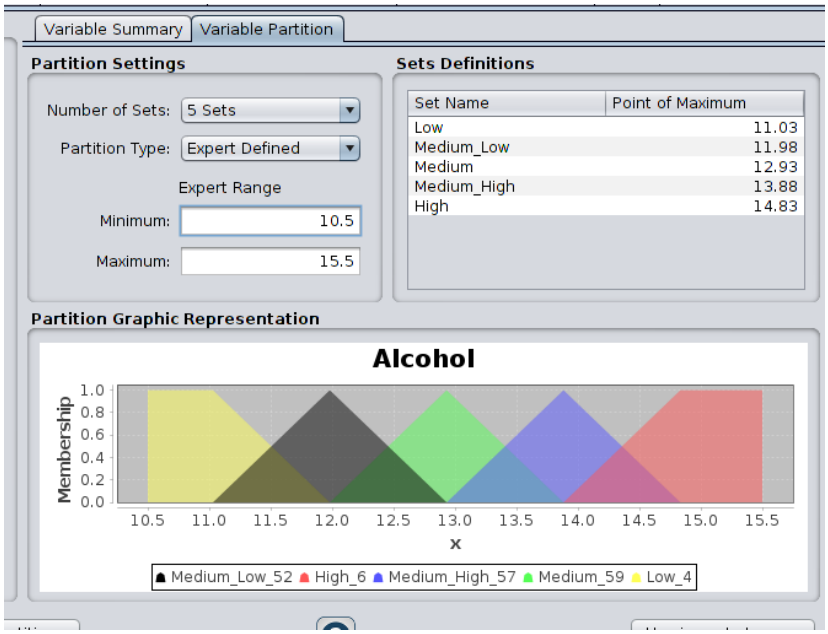


Figura 6.5: Aba da etapa de definição das partições *fuzzy*, mostrando a sub-aba de definição da partição da variável.

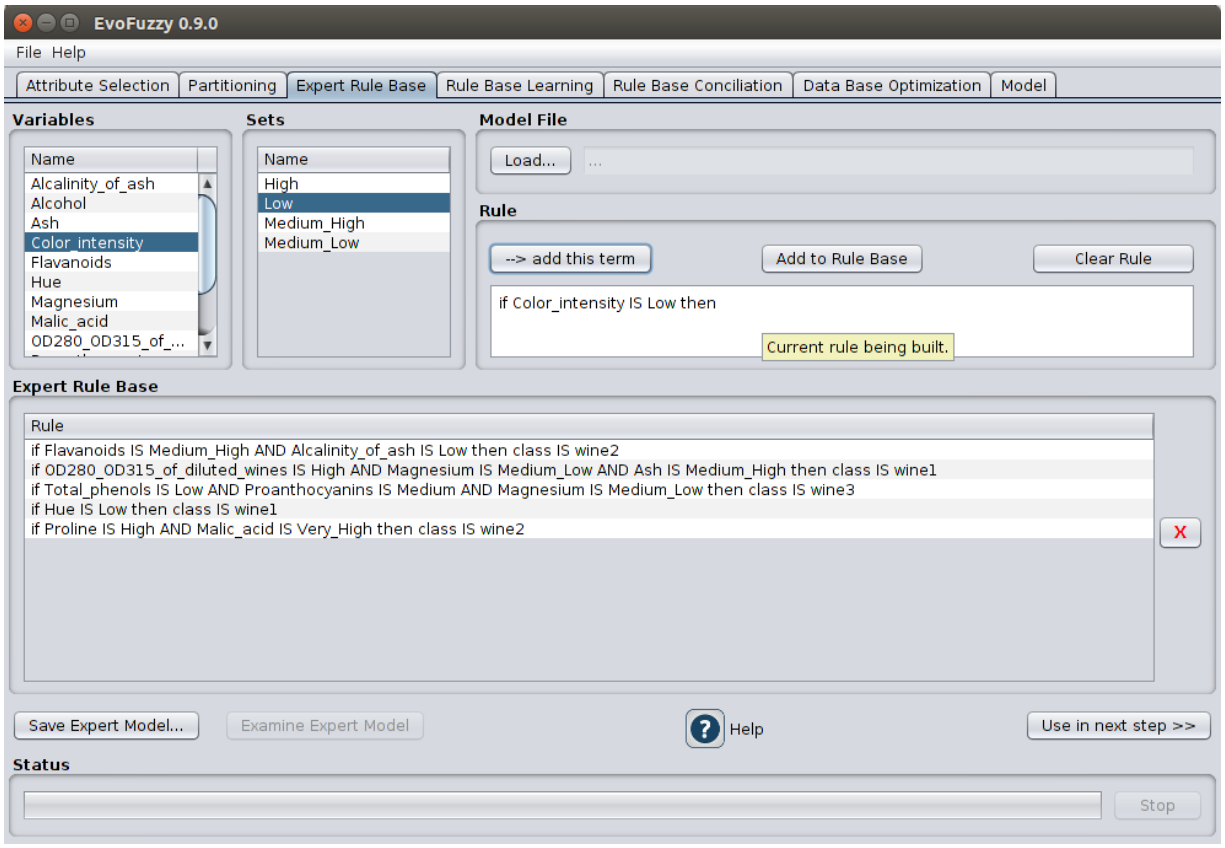


Figura 6.6: Aba da etapa de definição da BR Inicial, mostrando um exemplo de construção de uma BR.

A aba (*Rule Base Learning*) corresponde à etapa de aprendizado genético da BR (ver seção 5.4). Aqui o usuário deve definir os parâmetros de execução do algoritmo no painel *Settings* (Figura 6.8). Configurações avançadas podem ser definidas clicando-se no botão *Advanced* (Figura 6.7), entretanto, as configurações padrão já oferecem boa performance. Se o problema envolver muitas variáveis, podem ser necessárias mais gerações, já se ele for simples, poucas gerações são suficientes para alcançar resultados satisfatórios (no exemplo da Figura 6.7 foram usadas 100 gerações para o conjunto de dados *Wine* da UCI, que possui 13 variáveis e 178 instâncias).

Após o final da execução, as soluções encontradas são apresentadas na sub-aba *Front* e os resultados da validação cruzada no painel *Evaluation*. Selecionando uma solução, o usuário pode examinar o SFBR construído clicando no botão *'Examine Selected Model'*, o que o levará à aba *Model*. Adicionalmente, o resultado da validação agregada por número de regras nas soluções (ver seção 5.4.6) é apresentado na sub-aba *'Test Front Average'* (Figura 6.9). Gráficos mostrando a fronteira de Pareto formada no plano 'Interpretabilidade X erro' são mostrados na sub-aba *Graphic* (Figura 6.10).

O papel do especialista, então, é selecionar uma solução adequada na fronteira. Após isso, pode-se salvar um arquivo de modelo para uso posterior ou seguir para a próxima etapa clicando no botão *'Use in next step'*.

A aba (*Rule Base Conciliation*) corresponde à etapa de conciliação da BR (ver seção 5.5). Aqui o usuário (especialista) deve resolver conflitos provenientes da junção da BR definida pelo especialista e a BR aprendida pelo AGMO. É mostrado no painel *Conflict* (Figura 6.11), para cada conflito, sua natureza e as opções para solução. Após a solução de todos os conflitos, o usuário pode examinar o SFBR construído utilizando a BR conciliada clicando no botão *'Examine Conciliated Model'*, o que o levará à aba *Model*. Pode-se, também, salvar um arquivo de modelo para uso posterior ou seguir para a próxima etapa clicando no botão *'Use in next step'*.

A aba (*Data Base Optimization*) corresponde à etapa de otimização genética da BD (ver seção 5.6). Aqui o usuário deve definir os parâmetros de execução do algoritmo no painel *Settings* (Figura 6.12). Configurações avançadas podem ser definidas clicando-se no botão *Advanced* (uma janela semelhante à utilizada na etapa de aprendizado genético da BR é apresentada, como na Figura 6.7), entretanto, as configurações padrão já oferecem boa performance.

Após o final da execução, as soluções encontradas são apresentadas na sub-aba *Front* e os resultados da validação cruzada no painel *Evaluation*. Selecionando uma solução, o usuário pode examinar o SFBR construído clicando no botão *'Examine Selected Model'*, o que o levará

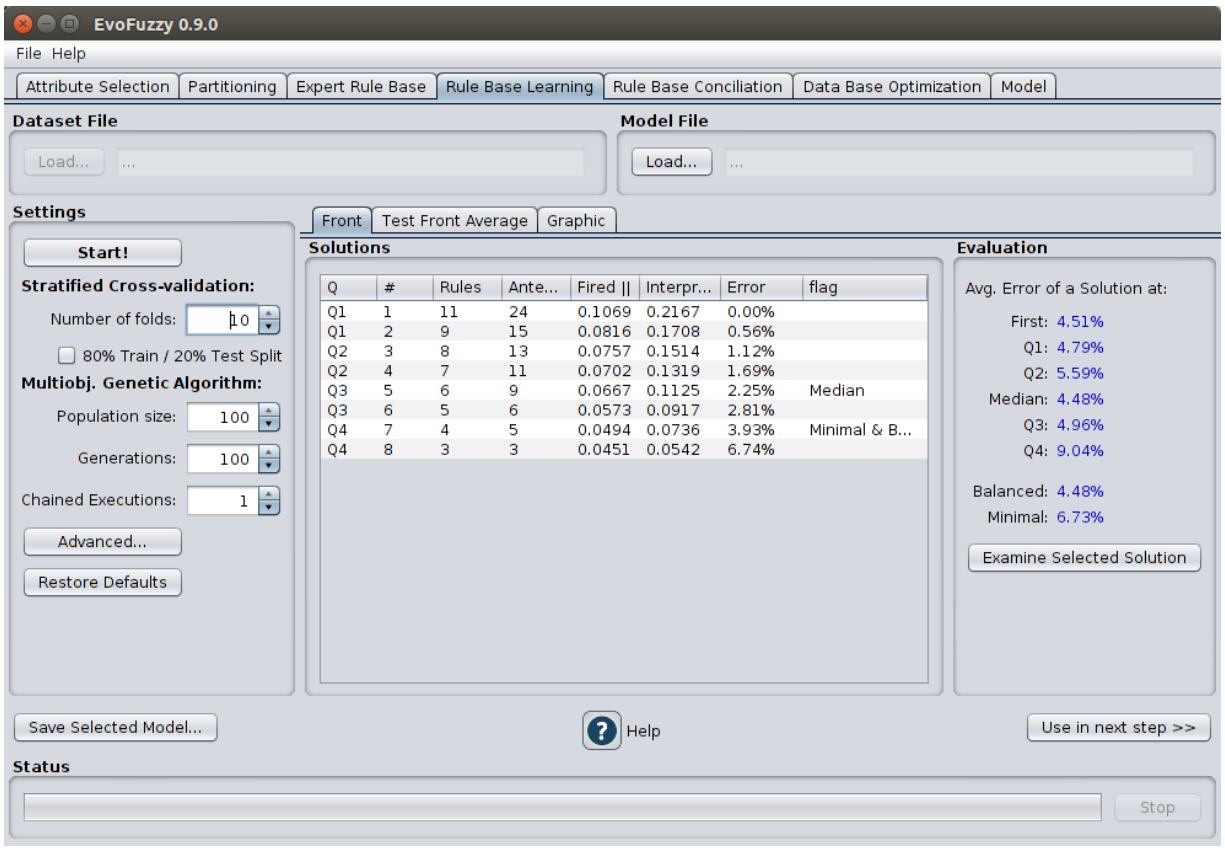


Figura 6.7: Aba da etapa de aprendizado genético da BR, mostrando a fronteira de Pareto resultante da execução na sub-aba *Front* e os resultados da validação cruzada no painel *Evaluation*.

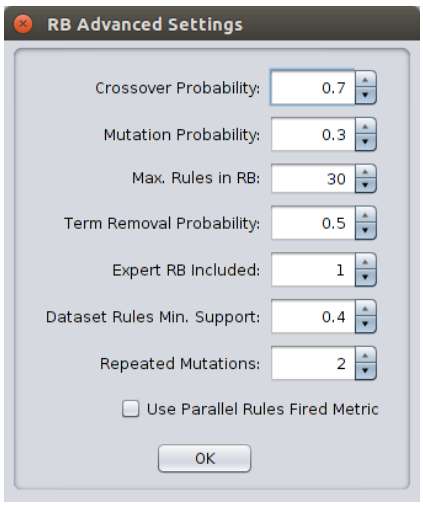
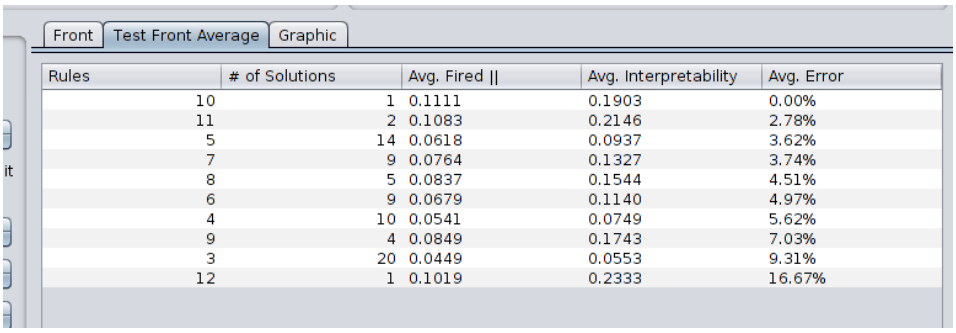


Figura 6.8: Janela de configurações avançadas da etapa de aprendizado genético da BR.



Rules	# of Solutions	Avg. Fired ||	Avg. Interpretability	Avg. Error
10	1	0.1111	0.1903	0.00%
11	2	0.1083	0.2146	2.78%
5	14	0.0618	0.0937	3.62%
7	9	0.0764	0.1327	3.74%
8	5	0.0837	0.1544	4.51%
6	9	0.0679	0.1140	4.97%
4	10	0.0541	0.0749	5.62%
9	4	0.0849	0.1743	7.03%
3	20	0.0449	0.0553	9.31%
12	1	0.1019	0.2333	16.67%

Figura 6.9: Aba da etapa de aprendizado genético da BR, mostrando a validação agregada por número de regras nas soluções na sub-aba *'Test Front Average'*.

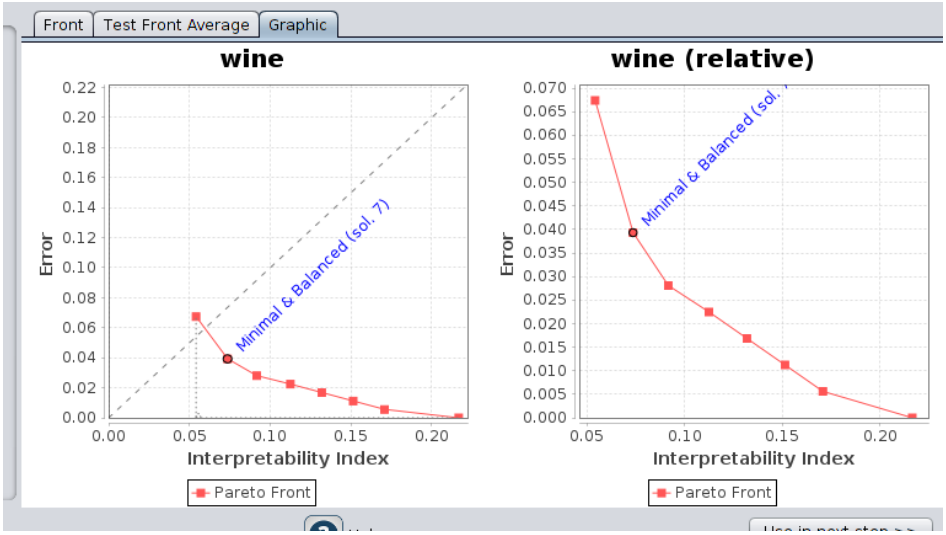


Figura 6.10: Aba da etapa de aprendizado genético da BR, mostrando gráficos da fronteira de Pareto formada no plano 'Interpretabilidade X erro' na sub-aba *Graphic*.

Figura 6.11: Aba da etapa de conciliação da BR, mostrando a solução de conflitos entre a BR definida pelo especialista e a BR aprendida pelo AGMO.

à aba *Model*. Adicionalmente, gráficos mostrando a fronteira de Pareto formada no plano 'Interpretabilidade X erro' são mostrados na sub-aba *Graphic*, semelhante aos utilizados na etapa de aprendizado genético da BR, como na Figura 6.10.

O papel do especialista, então, é selecionar uma solução adequada na fronteira. Após isso, pode-se salvar um arquivo de modelo para uso posterior ou inspecionar o modelo escolhido, que é o resultado final da abordagem, clicando no botão *'Use in next step'*.

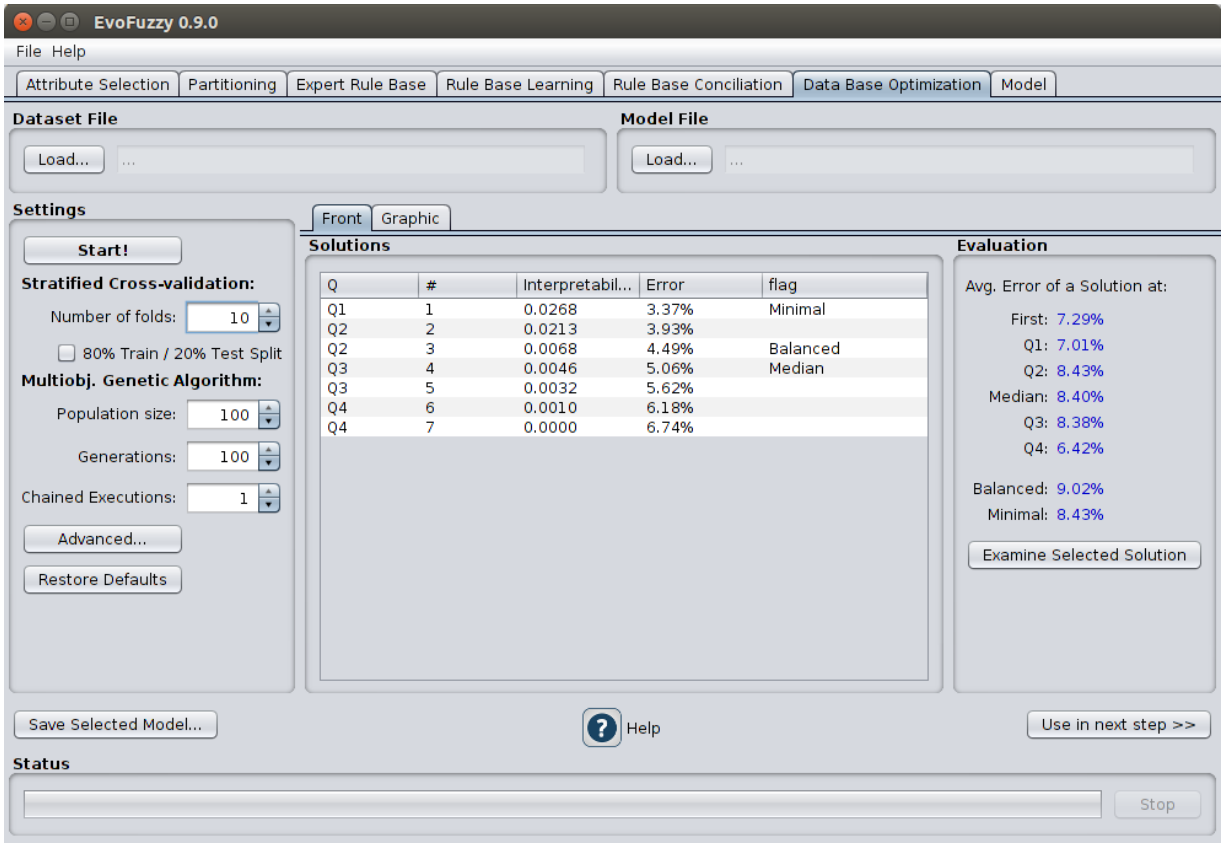


Figura 6.12: Aba da etapa de otimização genética da BR, mostrando a fronteira de Pareto resultante da execução na sub-aba *Front* e os resultados da validação cruzada no painel *Evaluation*.

A aba (*Model*) permite visualizar e interagir com os SFBR gerados durante as etapas do processo. Aqui o usuário pode efetuar inferências através da sub-aba *Inference* (Figura 6.13). Os valores das variáveis de entrada devem ser preenchidos no painel *'Input Variables'*; quando uma variável é selecionada, pode-se visualizar os CF e respectivos graus de pertinência no painel *'Sets Memberships'*. Quando todos os valores de entrada forem definidos, o resultado da inferência (seja um problema de classificação ou regressão) é apresentado no painel *Output*. Adicionalmente, as regras ativadas durante a inferência são apresentadas no painel *'Activated Rules'* e um gráfico da defuzzificação da variável de saída é mostrado no painel *'Defuzzification Graphic'*.

É possível inspecionar as partições *fuzzy* de cada variável da BD do SFBR através da sub-

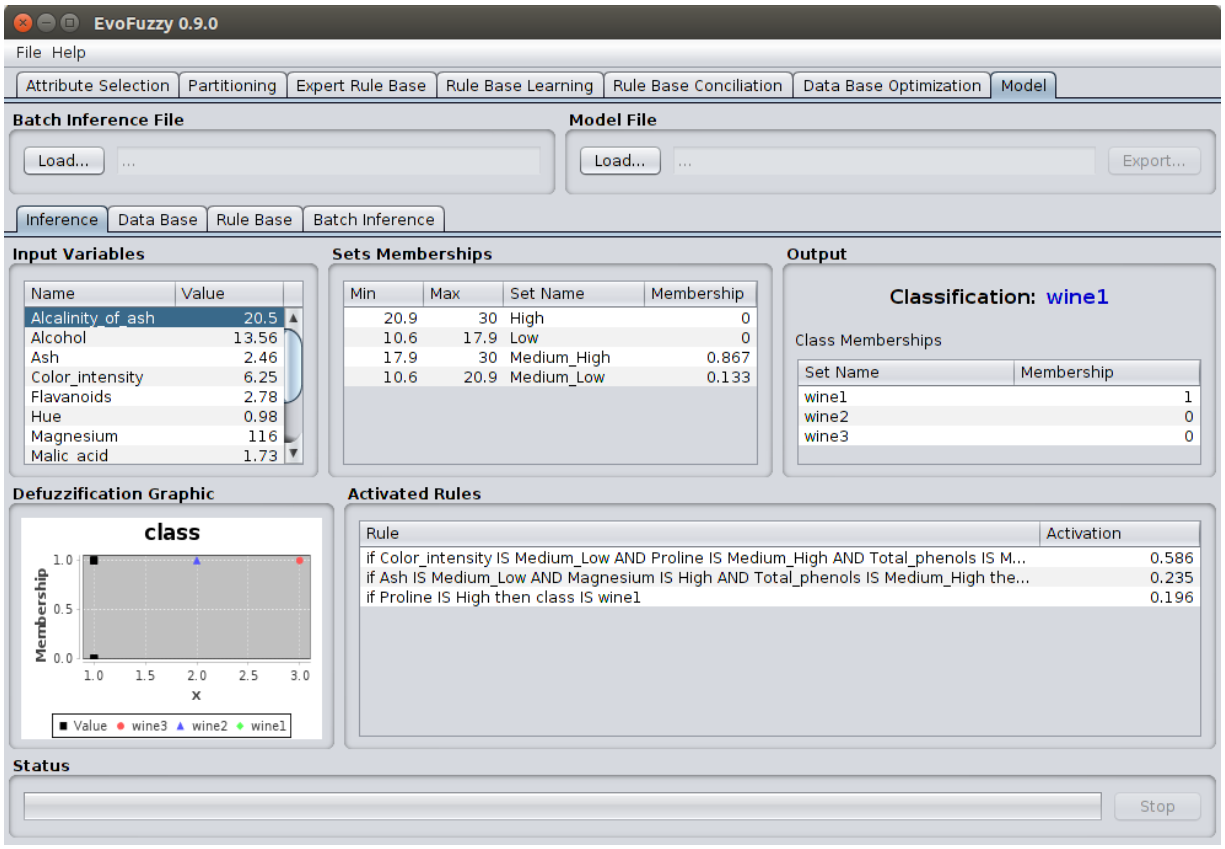


Figura 6.13: Aba *Model* que permite visualizar e interagir com um SFBR gerado pela ferramenta, mostrando os resultados de uma inferência na sub-aba *Inference*.

aba *'Data Base'* (Figura 6.14). Se uma inferência já foi efetuada, uma barra vertical assinala o valor de entrada definido para cada variável, permitindo visualizar o processo de fuzzificação. De maneira similar, é possível visualizar as regras que compõem a BR do SFBR através da sub-aba *'Rule Base'* (Figura 6.15).

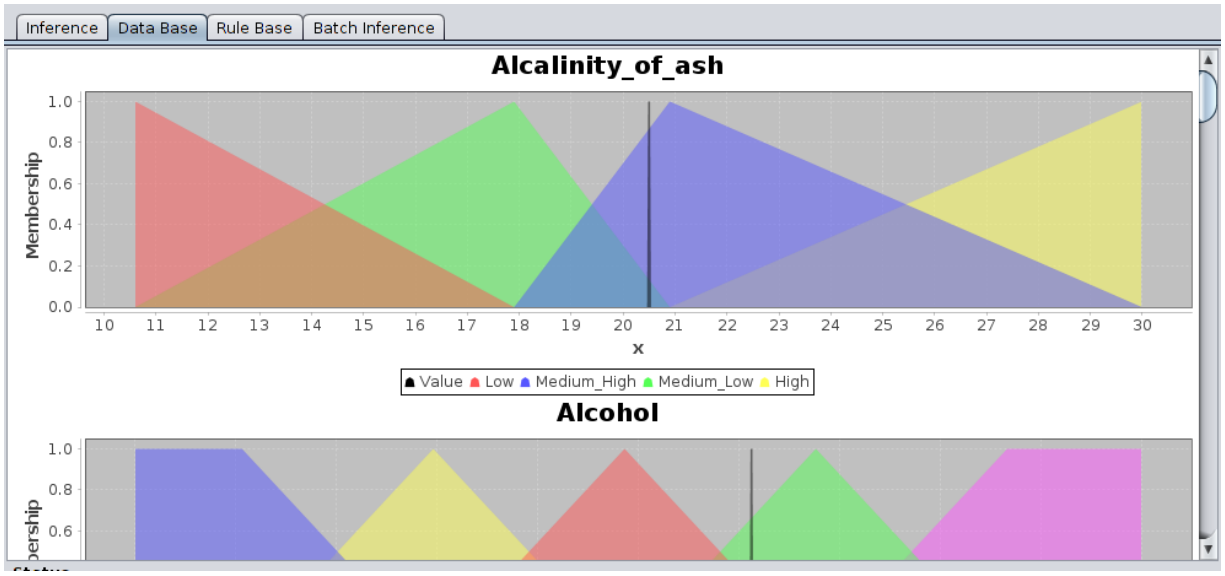


Figura 6.14: Aba *Model*, mostrando a partição de cada variável da BD do SFBR na sub-aba *'Data Base'* (a barra vertical corresponde ao valor de entrada da variável).

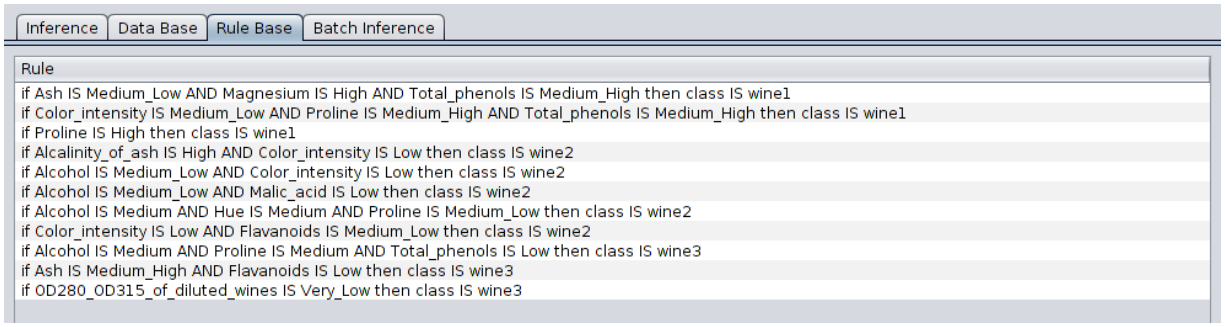


Figura 6.15: Aba *Model*, mostrando as regras da BR do SFBR na sub-aba *'Rule Base'*

Através da aba *'Batch Inference'* é possível inferir, de uma só vez, todas as instâncias de um conjunto de dados (Figura 6.16). Se este conjunto também possuir valores para a variável de saída, será apresentada a diferença de resultado entre os dados e o SFBR (se houve erro, em um problema de classificação, ou a diferença entre o valor esperado e o predito, no caso de uma regressão). Efetuando um duplo clique em qualquer instância da tabela, a inferência correspondente é apresentada na aba *Inference*. Por padrão, o conjunto de dados utilizado na construção do SFBR é carregado automaticamente, entretanto, um arquivo diferente pode ser carregado para inferência no painel *'Batch Inference File'*. De maneira semelhante, qualquer SFBR especificado em um arquivo FCL pode ser carregado para inspeção através do painel *'Model File'*.

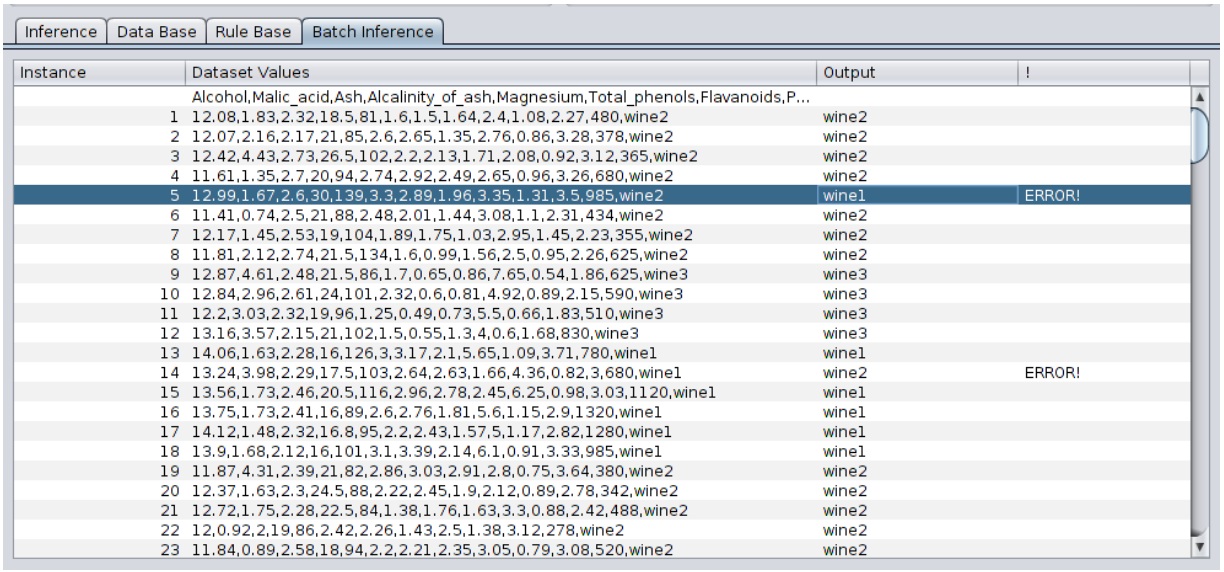


Figura 6.16: Aba *Model*, mostrando o resultado da inferência do SFBR sobre cada instância do conjunto de dados na sub-aba *Batch Inference*.

Capítulo 7

ESTUDO DE CASO

A bovinocultura é um dos principais ativos do agronegócio brasileiro e também um setor em constante evolução. Na busca por competitividade econômica e bem estar animal, a inseminação artificial[1] é uma das práticas de manejo que oferece melhor relação custo-benefício (cerca de 2% do custo total da produção). Entretanto, dados da Associação Brasileira de Inseminação Artificial indicam que apenas 11,9% do rebanho brasileiro utiliza-se de inseminação (ASBIA, 2014), sendo o restante das vacas cobertas por touros em monta natural.

Em regiões que utilizam pecuária extensiva, como o Pantanal Sul-Mato-Grossense, ainda existem dificuldades na introdução de técnicas mais sofisticadas de manejo, como a inseminação artificial. Nestes locais a utilização de touros para monta ainda é a principal técnica utilizada (NOGUEIRA et al., 2011).

Na reprodução de bovinos por monta natural, deseja-se que a cruza sempre resulte em prenhez da vaca, caso contrário, há perda econômica para o criador. Um dos mais importantes fatores de sucesso deste processo é a escolha criteriosa do touro. Selecionando-se animais com maior aptidão reprodutiva, a taxa de sucesso do processo é consideravelmente aumentada.

Uma das principais ferramentas utilizadas para diagnosticar a função reprodutiva do touro é o espermograma. Porém, segundo Nogueira et al. (2011), este pode apresentar alta variação decorrente de diversos fatores como ambiente, manejo nutricional, etc. Dessa forma, para que o exame andrológico seja eficiente como ferramenta de seleção de touros para a reprodução, deve ser realizado de forma completa, e somente quando várias característica são combinadas, torna-se possível identificar animais com melhor qualidade seminal.

Este processo depende de uma série de indicadores associados, ou não, entre si, indo de in-

[1] Entende-se por inseminação artificial a deposição mecânica do sêmen no aparelho reprodutivo da fêmea. Não confundir com fertilização in vitro. Na inseminação artificial, a fecundação, ou seja, a união do espermatozoide com o óvulo e a formação de um novo ser ocorrem naturalmente, sem a interferência do homem.

formações básicas, exame clínico e biometria testicular, além da avaliação dos aspectos físicos e morfológicos do sêmen, até os parâmetros de avaliações funcionais e bioquímicas (SALVADOR et al., 2008).

A análise destas informações para determinação da aptidão reprodutiva do animal constitui uma tarefa complexa, geralmente realizada por um especialista, logo, pode ser uma tarefa cara e pouco disponível ao público, sobretudo em regiões mais remotas como o Pantanal Sul-Mato-Grossense.

Algumas iniciativas para auxiliar nesta problemática existem na literatura, como a classificação andrológica por pontos (CAP) (DIAS et al., 2009), que baseia-se no somatório de alguns dos indicadores mencionados anteriormente. Entretanto, uma ferramenta capaz de captar as inter-relações entre estes indicadores e realizar uma análise multivariada de fato, ainda faz-se necessária.

O objetivo da modelagem descrita neste estudo de caso é construir um SFBR capaz de oferecer suporte à decisão durante a classificação da aptidão reprodutiva de touros Nelore criados de maneira extensivamente no Planalto e no Pantanal Sul-Mato-Grossense.

7.1 Material e Métodos

A metodologia adotada para a construção do SFBR foi a descrita no Capítulo 5, bem como a ferramenta EvoFuzzy foi utilizada para executar todos os seus passos. Desempenhando o papel de especialistas do domínio durante a modelagem, dois médicos veterinários e pesquisadores da Embrapa Pantanal[2] ficaram responsáveis pelas decisões.

7.1.1 Pré-processamento dos Dados

O conjunto de dados utilizado é fruto da avaliação andrológica de 4664 animais (NOGUEIRA et al., 2011), clinicamente sadios, mantidos exclusivamente a pasto com suplementação mineral durante o ano todo e pertencentes a diferentes propriedades nas regiões do Planalto e Pantanal Sul-Mato-Grossense. Foram coletadas 22 variáveis além da classe, constando de dados sobre o animal, exame clínico e exame do sêmen. Mais detalhes sobre esse tipo de avaliação podem ser vistos em Barbosa, Machado e Bergamaschi (2005). A classe é fornecida e corresponde à capacidade reprodutiva do animal, sendo considerado ***Apto***, ***Inapto*** ou ***Questi-***

[2] A Embrapa Pantanal é um das 47 unidades da Empresa Brasileira de Pesquisa Agropecuária – Embrapa, vinculada ao Ministério da Agricultura, Pecuária e Abastecimento. Sua área de atuação é a pesquisa para o desenvolvimento sustentável da agricultura na região do Pantanal.

onável. A classe *Questionável*, apesar de seu rótulo, é de interesse para a modelagem, pois, indica que o animal pode estar momentaneamente incapacitado, mas ainda não deve ser descartado como *Inapto* (por exemplo, um animal que ainda não atingiu a plena maturidade sexual). Não existem valores ausentes para nenhuma variável e os valores foram tratados durante a etapa de particionamento. Uma descrição do conjunto de dados é mostrada na Tabela 7.1.

Tabela 7.1: Descrição do conjunto de dados utilizado para modelagem.

#	Variável	Descrição	Tipo
1	regiao	Planalto ou Pantanal	Nominal
2	raca	Raça do animal	Nominal
3	genotipo	Genótipo do animal	Nominal
4	idade	Idade do animal (meses)	Numérico
5	classeid	Identificador de classe	Numérico
6	consis	Consistência testicular	Nominal
7	PE	Perímetro escrotal	Numérico
8	vol	Volume do ejaculado	Numérico
9	turbil	Turbilhonamento do sêmen	Numérico
10	motil	Motilidade do sêmen	Numérico
11	vigor	Vigor do sêmen	Numérico
12	acros	% defeitos acrossomo	Numérico
13	GCP	% defeitos gota citoplasmática proximal	Numérico
14	cabeca	% defeitos de cabeça	Numérico
15	pi	% defeitos peça intermediária	Numérico
16	maior	% defeitos maiores	Numérico
17	GCD	% defeitos gota citoplasmática distal	Numérico
18	isolnor	% cabeça isolada normal	Numérico
19	cauda	% defeitos de cauda	Numérico
20	menor	% defeitos menores	Numérico
21	anorma	% morfologicamente anormais	Numérico
22	norm	% morfologicamente normais	Numérico
23	conclusao (CLASSE)	Aptidão para reprodução	Nominal

Como pode ser visto no histograma da variável conclusão (classe do problema) apresentado na Figura 7.1, o conjunto de dados apresenta dois problemas inicialmente. O primeiro está relacionado à técnica utilizada pela abordagem proposta para aprendizado da BR e otimização da BD do SFBR, o AGMO. A quantidade de instâncias (4664) é muito grande e o processamento seria demasiadamente demorado, tornando-o inviável. O segundo problema, bem mais impactante, está relacionado à proporção desbalanceada de exemplos entre as classes. Note que a quantidade de exemplos para a classe *Apto* corresponde a aproximadamente 70% do total de exemplos.

Um conjunto de dados com classes desbalanceadas constitui um sério problema para a

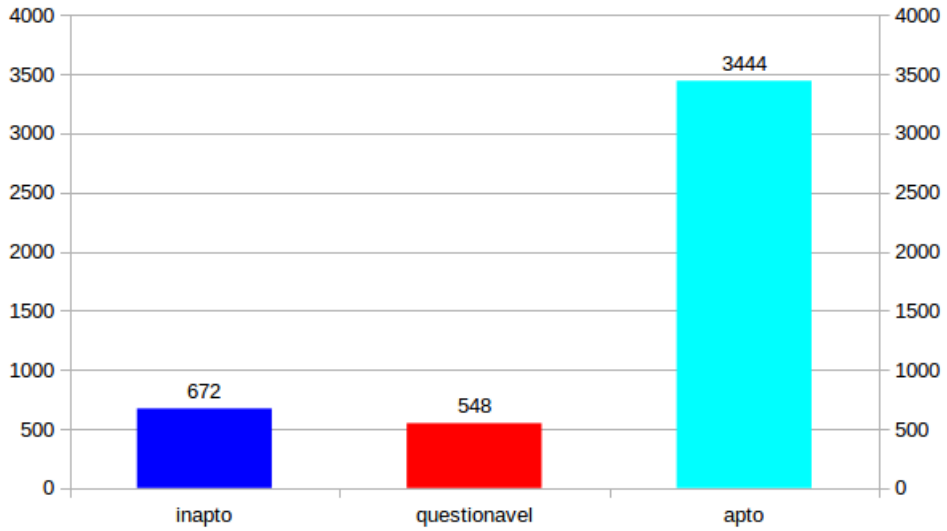


Figura 7.1: Histograma da variável conclusão (Classe) no conjunto de dados original.

maioria dos algoritmos de aprendizado de máquina para construção de classificadores, incluindo os adotados nesta abordagem. Isso se dá pela dificuldade que o algoritmo terá em aprender os conceitos relacionados às classes com muito menos exemplos no conjunto de dados. Neste contexto, os classificadores construídos tenderão a ter alta TCC para a classe predominante e TCC medíocre para as demais classes, gerando modelos de baixa qualidade.

Para contornar este problema, duas abordagens são frequentemente utilizadas, a sub-amostragem e a super-amostragem. Em Batista, Prati e Monard (2004) encontra-se um comparativo entre dez métodos para o tratamento de conjuntos de dados com classes desbalanceadas. Neste estudo de caso, como mencionado, o número de instâncias no conjunto de dados já é grande, de maneira que descartamos de imediato os métodos de super-amostragem.

Foi adotada a sub-amostragem randômica sem reposição para a classe majoritária (*Apto*). Um conjunto de teste estratificado contendo 20% das instâncias foi criado, com o restante das instâncias foram criadas duas amostras do conjunto de dados, uma contendo 45% do tamanho original, o que corresponde à um equilíbrio aproximado entre todas as classes; e outra contendo 75% do tamanho original, o que corresponde à uma redução de aproximadamente $2/3$ na quantidade de instâncias da classe *Apto*, mas ainda mantendo um bom número de exemplos para a classe.

Para testar qual dos percentuais de amostragem, a melhor opção seria utilizar o próprio AGMO implementado, entretanto isso seria inviável devido ao alto custo computacional. Como alternativa, foi realizado então um experimento simples, utilizando classificadores bem conhecidos na área e com vieses indutivos diversificados, à saber: ZeroR (sempre usa a classe majoritária como classificação), NaiveBayes, KNN (k=1 vizinho mais próximo), C4.5 e SVM.

Cada algoritmo foi executado 10 vezes e a média dos resultados foi comparada estatisticamente através do teste T pareado com valor de p=0.05. A medida de performance do classificador utilizada foi a Área sobre a curva ROC (WITTEN; FRANK; HALL, 2011). Dois resultados foram comparados; o primeiro tomou a medida somente para a classe *Questionável* (Tabela 7.2), visto esta ser a menos numerosa e com mais sobreposição de variáveis, sendo esta classe indicada pelos especialistas do domínio como a mais complicada de classificar; o segundo tomou a média ponderada da medida para todas as classes (Tabela 7.3).

Tabela 7.2: Comparativo da Área sobre a curva ROC para a classe *Questionável*.

Algoritmo	**Original**	**Amostra 45%**	**Amostra 70%**
ZeroR	0.50	0.50	0.50
NaiveBayes	0.83	0.80	0.81
KNN k=1	0.72	0.77 ○	0.77 ○
C4.5	0.81	0.81	0.82
SVM	0.71	0.77 ○	0.75 ○

○, • melhoria ou degradação estatisticamente significante

Tabela 7.3: Comparativo da média ponderada da Área sobre a curva ROC para todas as classes.

Algoritmo	**Original**	**Amostrado 45%**	**Amostrado 70%**
ZeroR	0.50	0.50	0.50
NaiveBayes	0.89	0.85 •	0.86 •
KNN k=1	0.81	0.81	0.82
C4.5	0.88	0.86	0.87
SVM	0.80	0.84 ○	0.83 ○

○, • melhoria ou degradação estatisticamente significante

Observando-se os resultados das Tabelas 7.2 e 7.3, nota-se que, para grande maioria dos algoritmos, não houve degradação da performance para ambas os percentuais de amostragem, sendo que para alguns algoritmos, houve melhoria da performance. Observa-se também que ambos os percentuais se equivalem, com ligeira vantagem para a amostra à 70%. Dito isto, nossa opção foi utilizar esta última, visto amenizar o problema de desbalanceamento de classes mas ainda mantendo uma boa quantidade de exemplos para a classe *Apto*. Foi criado então uma amostra à 70% do conjunto de dados original, que foi usada nos passos subsequentes. A distribuição das classes no conjunto amostrado pode ser vista na Figura 7.2.

7.1.2 Seleção de atributos

Realizado o pré-processamento do conjunto de dados, iniciou-se a sequencia de passos descrita no Capítulo 5, cujo primeiro constitui a seleção de atributos. Os dados foram submetidos

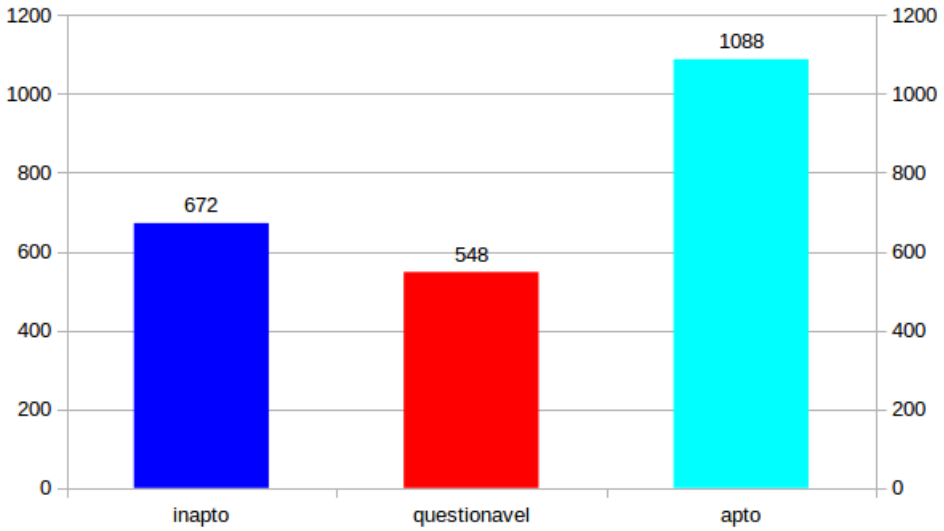


Figura 7.2: Histograma da variável conclusão (Classe) no conjunto de dados amostrado à 70%.

à EvoFuzzy e os resultados do processo podem ser vistos na Tabela 7.4.

Observando os resultados da seleção dos atributos relevantes por parte do especialista (denotada por @ na Tabela 7.4), pode-se inferir algumas constatações, que discutiremos a seguir.

Devido à quantidade de variáveis do conjunto de dados (23), o papel do especialista é ainda mais importante, visto que, por melhores que sejam os algoritmos de seleção de atributos, o especialista é ainda quem melhor entende o problema e como os dados foram gerados. Dois exemplos disso podemos destacar: o atributo idade, cuja sugestão do processo seria ser descartado, é citado como um dos mais importantes para o manejo dos touros, sendo o principal fator determinante quando não é possível realizar um exame andrológico completo (NOGUEIRA et al., 2011); além disso, todos os atributos relacionados com defeitos nos espermatozoides detectados através do exame de microscopia do sêmen (*cauda*, *isolnor*, *acros*, *GCP*, *pi*, *cabeca* e *GCD*), foram descartados pelos especialistas, pois são utilizados para cálculo dos atributos *norm*, *anorma*, *maior* e *menor*, muito mais importantes para a avaliação. Portanto, mantê-los apenas adicionaria redundância ao modelo, porém, não saberíamos disso apenas olhando para os dados, já que tais atributos tem alta correlação com a classe, como mostrado pelas suas posições no ranqueamento.

Observa-se também que o método proposto, que baseia-se na junção de vários algoritmos, mostrou-se uma ferramenta valiosa para auxiliar o especialista quanto à seleção de atributos. Note que os 6 primeiros atributos escolhidos pelo especialista coincidem com as 6 primeiras posições do ranking e com os 6 atributos selecionados pelo algoritmo CFS.

Tabela 7.4: Resultado do processo de seleção de atributos sobre o conjunto de dados do estudo de caso.

Recomendacao	Ranking(stdev)	Sel.	Atributo		
Manter	1.7 +– 2.53	* t	22 norm	@	
Manter	2.2 +– 2.52	* t	21 anorma	@	
Manter	2.8 +– 2.23	* t	16 maior	@	
Manter	4.7 +– 2.34	*	20 menor	@	
Manter	5.8 +– 1.97	* t	6 consis	@	
Manter	6.2 +– 1.92	* t	10 motil	@	
Manter	6.7 +– 2.39	t	19 cauda		
Manter	10.7 +– 1.74		18 isolnor		
Manter	11.4 +– 2.46		12 acros		
Manter	11.9 +– 2.36		8 vol		
Manter	12.8 +– 2.08		13 GCP		
Manter	13.8 +– 1.77		2 raca		
					50%
Manter	13.9 +– 1.67	t	7 PE	@	
Manter	14.4 +– 1.46	t	15 pi		
Descartar	14.8 +– 2.06		14 cabeca		
Manter	15 +– 1.63	t	1 regiao		
Descartar	15.3 +– 3.38		4 idade	@	
					70%
Descartar	15.7 +– 2.12		11 vigor	@	
Descartar	17.6 +– 1.99		17 GCD		
Descartar	17.7 +– 1.27		5 classeid		
Manter	18.1 +– 1.21	t	9 turbil		
Descartar	19.7 +– 1.18		3 genotipo	@	

* = CFS t = C4.5 @ = escolha do especialista

7.1.3 Definição das partições *fuzzy*

Esta possivelmente é a etapa onde o especialista tenha o papel mais importante. No contexto de sistemas de suporte à decisão, a semântica das partições é um elemento chave para que o resultado seja o esperado. A EvoFuzzy ofereceu subsídio ao entendimento dos dados (ver seção 5.2), bem como o ferramental para definição das partições. Métodos para o particionamento automatizado são muito interessantes quando há pouco conhecimento acerca de uma variável, mas neste estudo de caso, devido o conhecimento consolidado que os especialista possuem acerca das variáveis selecionadas, eles optaram por definir manualmente as partições de todas as variáveis numéricas. O resultado deste processo pode ser visto nas Figuras 7.3 e 7.4.

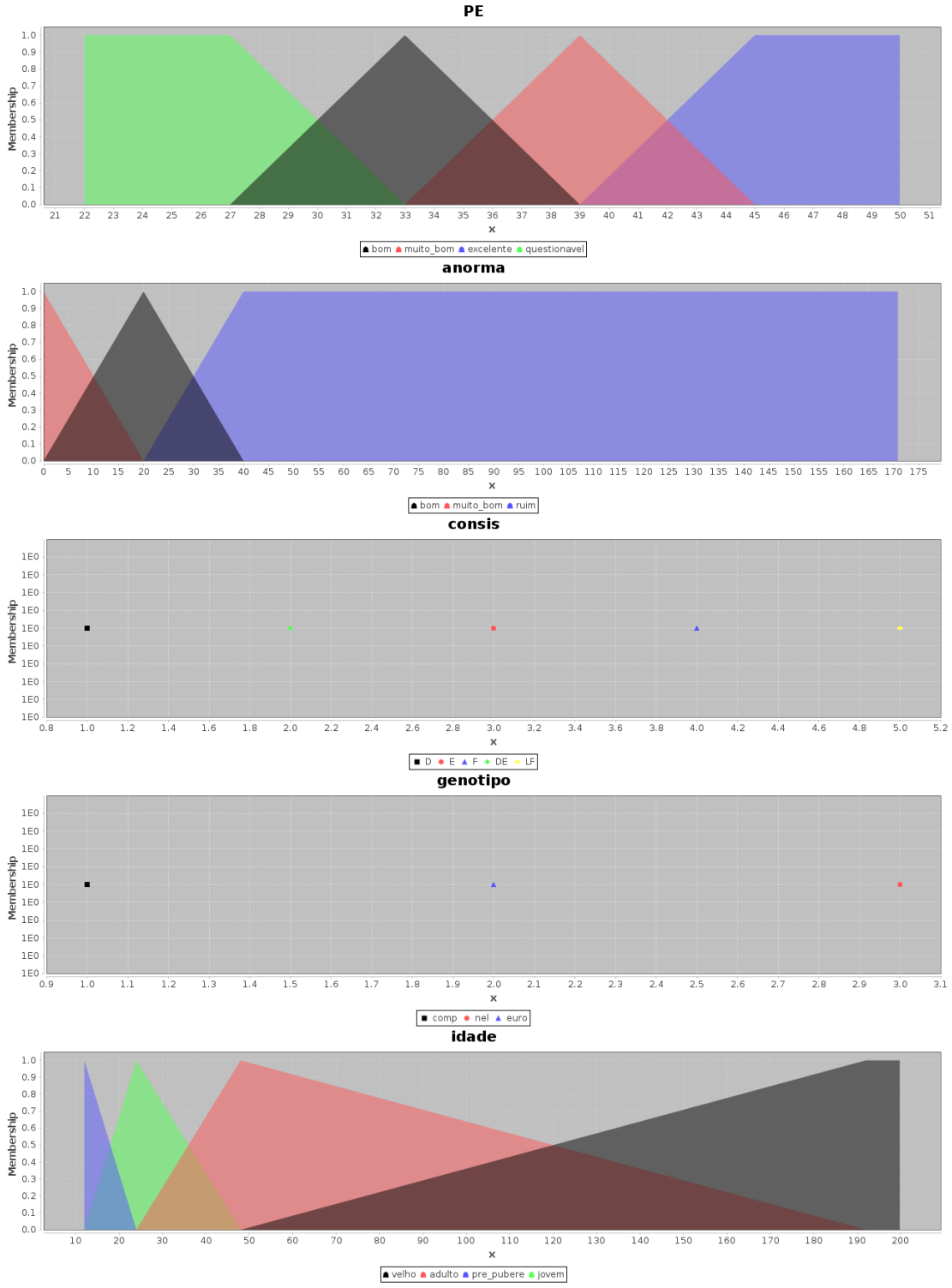


Figura 7.3: Particionamento das variáveis *PE*, *anorma*, *consis*, *genotipo* e *idade*, definido pelos especialistas com auxílio da EvoFuzzy.

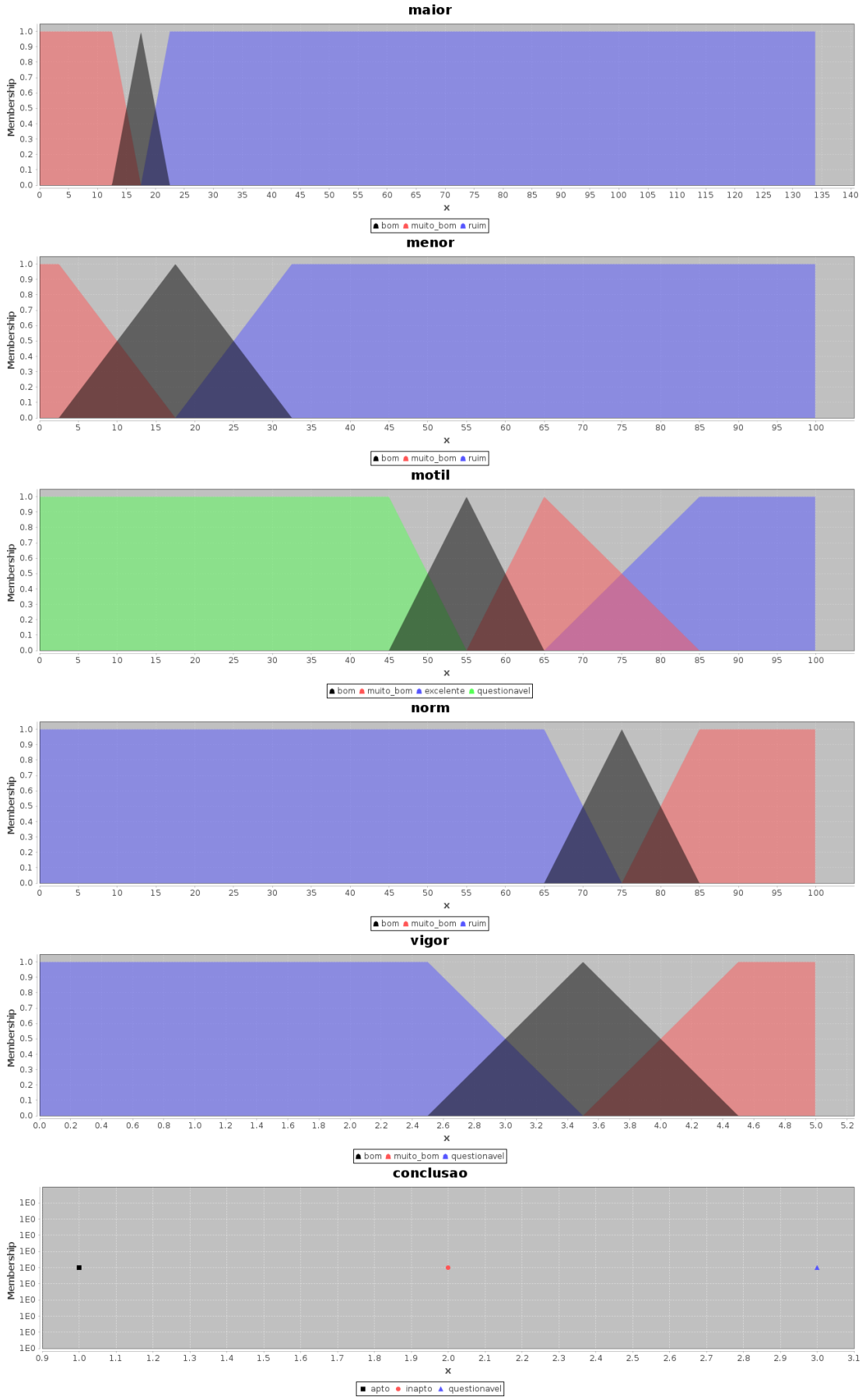


Figura 7.4: Particionamento das variáveis *maior*, *menor*, *motil*, *norm*, *vigor* e *conclusão*, definido pelos especialistas com auxílio da EvoFuzzy.

7.1.4 Definição da base de regras inicial

Assim como na etapa anterior, a definição das RE é essencial para que o aprendizado da BR executado pelo AGMO seja direcionado ao resultado esperado. Utilizando a EvoFuzzy para a definição das regras, os especialistas definiram 14 regras, apresentadas na Tabela 7.5.

Tabela 7.5: Base de regras inicial definida pelos especialistas.

1	**SE** *PE* **É** *questionavel* **E** *idade* **É** *adulto* **ENTÃO** *conclusao* **É** *inapto*
2	**SE** *vigor* **É** *questionavel* **ENTÃO** *conclusao* **É** *questionavel*
3	**SE** *idade* **É** *jovem* **E** *anorma* **É** *ruim* **ENTÃO** *conclusao* **É** *questionavel*
4	**SE** *idade* **É** *adulto* **E** *anorma* **É** *ruim* **ENTÃO** *conclusao* **É** *inapto*
5	**SE** *PE* **É** *bom* **E** *norm* **É** *ruim* **ENTÃO** *conclusao* **É** *questionavel*
6	**SE** *motil* **É** *questionavel* **ENTÃO** *conclusao* **É** *questionavel*
7	**SE** *motil* **É** *questionavel* **E** *norm* **É** *ruim* **E** *PE* **É** *questionavel* **ENTÃO** *conclusao* **É** *inapto*
8	**SE** *maior* **É** *ruim* **ENTÃO** *conclusao* **É** *questionavel*
9	**SE** *menor* **É** *ruim* **ENTÃO** *conclusao* **É** *questionavel*
10	**SE** *consis* **É** *F* **E** *motil* **É** *questionavel* **E** *maior* **É** *ruim* **ENTÃO** *conclusao* **É** *inapto*
11	**SE** *consis* **É** *F* **E** *anorma* **É** *ruim* **ENTÃO** *conclusao* **É** *questionavel*
12	**SE** *PE* **É** *bom* **ENTÃO** *conclusao* **É** *apto*
13	**SE** *motil* **É** *bom* **E** *norm* **É** *bom* **ENTÃO** *conclusao* **É** *apto*
14	**SE** *anorma* **É** *bom* **ENTÃO** *conclusao* **É** *apto*

7.1.5 Aprendizado Genético da BR

Definidas a BD e a BR iniciais, o processo de aprendizado genético da BR foi executado. Os parâmetros definidos para o algoritmo foram: tamanho da população = 100, tamanho máximo da base de regras = 30 e quantidade de gerações = 1500. Para o restante das configurações foram mantidos os valores padrão da EvoFuzzy. Foi utilizada validação cruzada estratificada de 10 partições. A fronteira de Pareto resultante da execução pode ser vista na Tabela 7.6 enquanto o resultado da validação cruzada é mostrado na Tabela 7.7.

Examinado os resultados da validação, é possível verificar que as soluções que apresentaram menos erro foram as compreendidas entre a primeira solução e a solução mediana, ou seja, as soluções do primeiro (Q1) e segundo (Q2) quartis da fronteira de Pareto.

Os especialistas puderam inspecionar os detalhes e realizar inferências nos SFBR obtidos através da aba *Model* da Evofuzzy, focando a busca no intervalo de soluções mais promissor. Ao final, julgaram que a solução número 9 foi a mais adequada segundo seus conhecimentos do domínio. Esta solução também é uma das que se posiciona em um ponto na fronteira onde

o aumento de regras na solução resulta em muito pouco aumento da acurácia. A BR codificada nesta solução pode ser vista na Tabela 7.8.

É importante ressaltar que se a escolha da solução adequada tivesse sido realizada de maneira automática e não pelo especialista, como em várias outras abordagens que utilizam aprendizado multiobjetivo para construção de SFBR, 34 das 35 soluções não dominadas encontradas teriam sido ignoradas neste processo, podendo resultar que alguma solução mais adequada nem sequer fosse analisada.

7.1.6 Conciliação da base de regras

Neste etapa as BR do especialista (Tabela 7.5) e aprendida pelo AGMO (Tabela 7.8) foram submetidas à EvoFuzzy para análise dos conflitos lógicos resultantes da junção delas. A Tabela 7.9 apresenta a saída de console da ferramenta para esta análise. Através do ambiente interativo oferecido, onde são mostrados os suportes e a alteração no erro resultante de cada ação (ver exemplo da Figura 6.11), os especialistas puderam avaliar cada situação conflitante e efetuar a conciliação. O resultado deste processo foi a BR final do SFBR, apresentada na Tabela 7.10, que alia regras do especialista, consideradas essenciais para manutenção da semântica, às regras aprendidas dos dados, que contribuem para o aumento da acurácia do modelo.

7.1.7 Otimização Genética da Base de Dados

O SFBR formado pela BD definida pelo especialista e a BR conciliada, foi submetido à etapa de otimização genética da BD. Os parâmetros definidos para o algoritmo foram: tamanho da população = 100 e quantidade de gerações = 1500. Para o restante das configurações foram mantidos os valores padrão da EvoFuzzy. Foi utilizada validação cruzada estratificada de 10 partições. A fronteira de Pareto resultante da execução pode ser vista na Tabela 7.11 enquanto o resultado da validação cruzada é mostrado na Tabela 7.12.

Examinado os resultados da validação, é possível verificar que as soluções que apresentaram menos erro foram as do primeiro quartil (Q1) da fronteira de Pareto, sendo a solução mínima a que teve melhor desempenho entre estas.

Os especialistas puderam inspecionar os detalhes e realizar inferências nos SFBR obtidos através da aba *Model* da Evofuzzy, focando a busca em Q1. Ao final, julgaram que a solução número 6 (mínima) é a mais adequada, visto que apresenta pouca alteração no formato dos CF em relação a BD original enquanto oferece a melhor acurácia. As partições otimizadas codificadas por esta solução podem ser vistas na Figura 7.5.

A mesma situação problemática, discutida na etapa de aprendizado genético da BR, quanto a escolha de uma solução na fronteira de Pareto, se aplica à este passo também. A escolha realizada de maneira automática e não pelo especialista, implica no desperdício de possíveis soluções adequadas. Além disso, somente um especialista é capaz de determinar se a semântica das partições foi demasiadamente alterada durante o processo de otimização, já que este é um conceito intangível.

O código fonte em linguagem FCL para o SFBR construído através da aplicação da abordagem pode ser visto no Apêndice A.1.

7.2 Resultados e Discussão

Para avaliarmos o resultado alcançado através da aplicação da abordagem proposta, alguns aspectos precisam ser observados. Quanto à semântica correta, esta foi garantida pela definição dos componentes do modelo (BR e BD) e, também, supervisão durante todas as etapas do processo, pelos especialistas do domínio. Nada mais há o que falar senão que os especialistas acataram o modelo.

Quanto à acurácia, foi realizado um experimento que buscou descobrir se o SFBR construído teria performance equivalente à outros algoritmos clássicos da área. Para tal, foram utilizados novamente os algoritmos ZeroR, NaiveBayes, KNN (k=1), C4.5 e SVM. Foi adotada validação cruzada de 10 partições e cada algoritmo foi executado 10 vezes, sendo calculada a média do erro na classificação. Devido o processo de validação cruzada em algoritmos multiobjetivo não permitir que o erro de uma solução específica seja estimado (no nosso caso, a solução escolhida pelos especialistas como solução final), o resultado foi comparado ao erro médio das soluções do primeiro quartil da fronteira (à qual pertence a solução escolhida), podendo ser visto na Tabela 7.13.

Como a abordagem proposta constrói o SFBR de maneira interativa com o especialista, não é possível repetir o processo e, portanto, não é possível realizar um teste estatístico para determinar se as diferenças na acurácia em relação aos algoritmos clássicos são relevantes. Porém, observando os resultados, pode-se deduzir que a acurácia da abordagem proposta é, pelo menos, equivalente à estes classificadores, já que obteve o menor erro entre todos no comparativo.

Falando especificamente quanto à metodologia criada, realizamos outro experimento que buscou descobrir se a integração entre o conhecimento do especialista e o induzido dos dados alcança os objetivos propostos. Para isso, dois outros SFBR foram criados. O primeiro foi construído de maneira completamente automática a partir dos dados, sem nenhuma interferência do

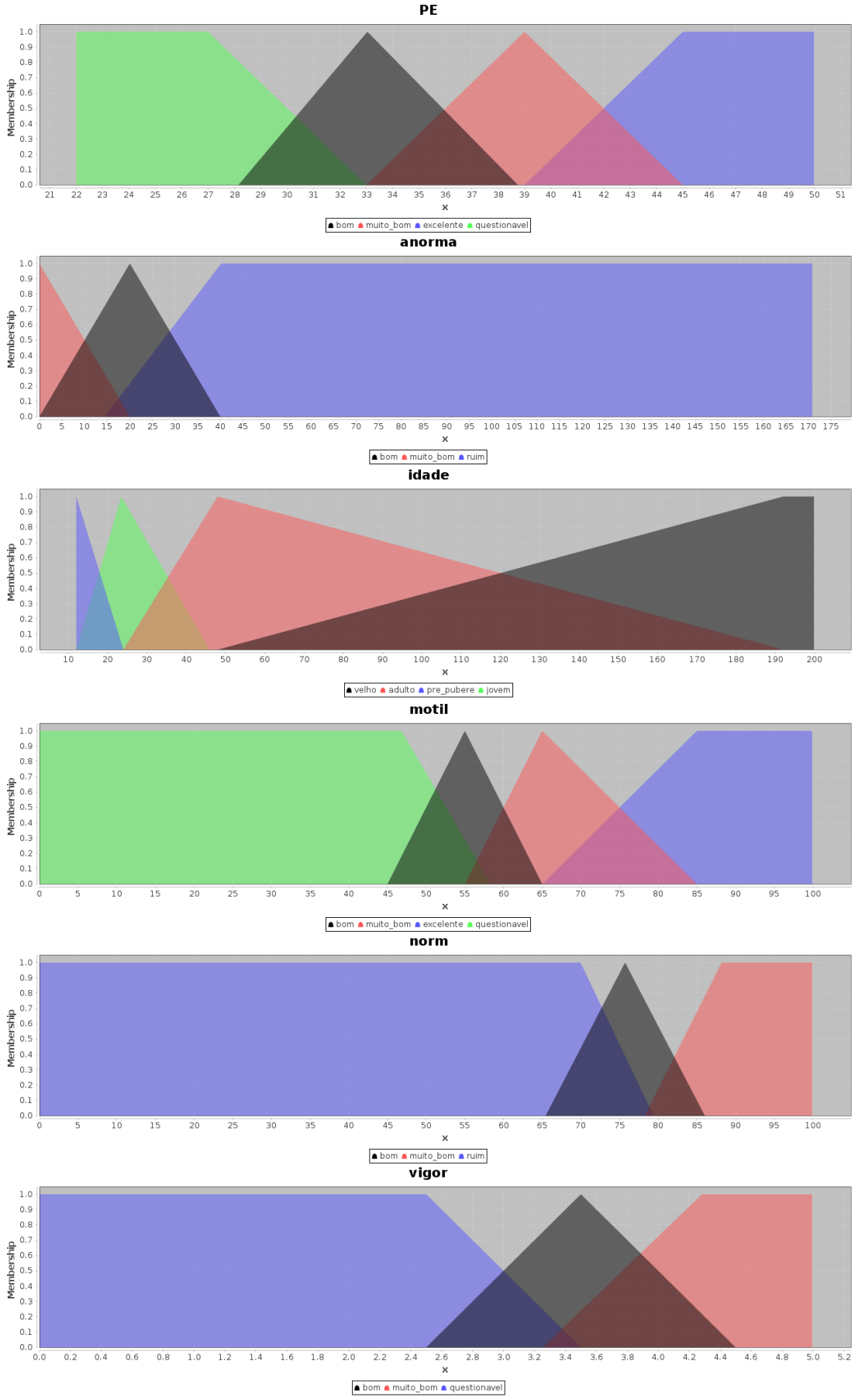


Figura 7.5: Particionamento das variáveis *PE*, *anorma*, *idade*, *motil*, *norm* e *vigor*, modificadas pelo processo de otimização genética da BD.

especialista. Para isso, também foi utilizada a EvoFuzzy. Foi simulado que o especialista não sabia definir nenhum componente do modelo, como a seguir: na etapa de seleção de atributos, acatou-se todas as recomendações automáticas da ferramenta; no particionamento, todas as variáveis tiveram seus parâmetros descobertos automaticamente; não foi adicionada nenhuma regra do especialista e, portanto, o aprendizado genético não foi direcionado e nem houve conciliação de BR. Para escolha da solução final, foi tomada a solução mediana na fronteira de Pareto, visto ser este o critério adotado por boa parte das abordagens automáticas para construção de SFBR (ISHIBUCHI; NOJIMA, 2013a). O código fonte em linguagem FCL para este SFBR pode ser visto no Apêndice A.2.

O segundo SFBR, foi construído de maneira completamente manual pelos especialistas. Esta tarefa foi realizada antes da modelagem do estudo de caso, pois não desejávamos que após utilizar a ferramenta para a construção, suas decisões ficassem enviesadas pelos subsídios que ela oferece. Foi solicitado aos especialistas que cumprissem os mesmos passos básicos, selecionar atributos, definir suas partições *fuzzy* e definir a base de regras.

Como mencionado anteriormente, formalizar seu conhecimento acerca de um domínio não é uma tarefa trivial para o especialista, sobretudo em um fenômeno complexo como o que é objeto do estudo de caso, possuindo 22 variáveis de entrada e 3 classes possíveis como saída. É muito mais fácil para o especialista julgar um cenário apresentado segundo seu *expertise* do que descrever o funcionamento do fenômeno de maneira detalhada. Não surpreendentemente, a abordagem escolhida pelos especialistas para a construção manual foi, selecionar um pequeno grupo de variáveis (*PE*, *anorma*, *idade*, *motil* e *vigor*) e construir a base de regras enumerando as combinações possíveis entre os CF Definidos para cada variável, gerando uma base de regras extensa e pouco interpretável. O código fonte em linguagem FCL para este SFBR pode ser visto no Apêndice A.3.

Foram contabilizados alguns indicadores da complexidade estrutural dos componentes (número de regras, Antecedentes das regras, Variáveis e CF) para cada um dos modelos. Quanto à acurácia, para a abordagem proposta e a construção automática, foram comparados os erros médios na classificação das soluções em Q1 obtidos via validação cruzada estratificada de 10 partições; para a construção manual, foi aferido o erro na classificação sobre o conjunto de treinamento. Os dados do comparativo entre os métodos para construção de SFBR podem ser vistos na Tabela 7.14.

Observando os resultados do comparativo, fica evidente que a construção de SFBR de maneira inteiramente manual é uma tarefa complexa, que demanda muito tempo e exige muito critério para que seja alcançado um resultado satisfatório. No comparativo, este método de

construção foi o que resultou no SFBR com maior complexidade, o que implica em baixa interpretabilidade, mesmo tendo sido definido pelo especialista. Além disso, foi o que apresentou o maior erro ($51,9\%$), muito próximo do erro da classe majoritária ($52,8596\%$), o que nos indica um péssimo classificador.

Quanto a abordagem proposta e a construção automática, em termos de acurácia, a diferença de apenas $0,52\%$ no erro médio na classificação pode ser considerada desprezível e é mais que compensada pela menor complexidade estrutural obtida pela abordagem proposta.

A questão chave neste comparativo, é que apesar dos resultados serem semelhantes para os dois métodos, a abordagem proposta propiciou a semântica esperada pelo especialista, enquanto abordagens completamente automáticas dificilmente teriam alcançado esse objetivo.

Tabela 7.6: Fronteira de Pareto resultante da execução do aprendizado genético da BR.

Q	#	Regras	Antecedentes	Interpretabilidade	Erro na classificação	
	1	18	33	0.3550	0.2188	
	2	17	32	0.3366	0.2192	
	3	17	31	0.3350	0.2196	
	4	16	30	0.3166	0.2201	
Q1	5	16	29	0.3150	0.2209	
	6	15	28	0.2966	0.2214	
	7	15	27	0.2950	0.2227	
	8	14	26	0.2766	0.2231	
	9	14	25	0.2750	0.2244	
	10	13	24	0.2566	0.2253	
	11	13	23	0.2550	0.2257	
	12	13	22	0.2533	0.2274	
	13	12	21	0.2350	0.2279	
Q2	14	11	20	0.2166	0.2309	
	15	11	19	0.2150	0.2313	
	16	10	18	0.1966	0.2339	
	17	10	17	0.1950	0.2352	
	18	10	16	0.1933	0.2374	Med
	19	9	16	0.1766	0.2378	
	20	9	15	0.1750	0.2387	
	21	9	14	0.1733	0.2409	
	22	8	13	0.1550	0.2426	
Q3	23	8	12	0.1533	0.2448	
	24	7	11	0.1350	0.2482	
	25	7	10	0.1333	0.2543	
	26	6	10	0.1166	0.2551	Min
	27	6	9	0.1150	0.2569	
	28	6	8	0.1133	0.2634	
	29	5	8	0.0966	0.2642	Bal
	30	5	7	0.0950	0.2707	
Q4	31	5	6	0.0933	0.2876	
	32	4	6	0.0766	0.2937	
	33	4	5	0.0750	0.2985	
	34	3	5	0.0583	0.3149	
	35	3	4	0.0566	0.3184	

Med = Solução mediana, Min = Solução mínima, Bal = Solução balanceada.

Tabela 7.7: Resultado da validação cruzada estratificada de 10 partições para o processo de aprendizado genético da BR.

Posição na Fronteira	Erro médio na classificação
Solução balanceada	27.0794%
Solução mínima	26.9923%
Solução mediana	24.4352%
Primeira solução	24.6081%
Média das soluções em Q1	24.4490%
Média das soluções em Q2	24.6729%
Média das soluções em Q3	25.3378%
Média das soluções em Q4	28.3980%

Tabela 7.8: Base de regras codificada pela solução 9 da fronteira de Pareto resultante do aprendizado genético da BR.

1	**SE** *cons* **É** *DE* **ENTÃO** *conclusao* **É** *apto*
2	**SE** *idade* **É** *adulto* **E** *menor* **É** *muito_bom* **ENTÃO** *conclusao* **É** *apto*
3	**SE** *norm* **É** *muito_bom* **ENTÃO** *conclusao* **É** *apto*
4	**SE** *PE* **É** *questionavel* **ENTÃO** *conclusao* **É** *inapto*
5	**SE** *anorma* **É** *ruim* **ENTÃO** *conclusao* **É** *inapto*
6	**SE** *cons* **É É** *LF* **E** *idade* **É** *jovem* **ENTÃO** *conclusao* **É** *inapto*
7	**SE** *idade* **É** *adulto* **E** *motil* **É** *questionavel* **ENTÃO** *conclusao* **É** *inapto*
8	**SE** *norm* **É** *ruim* **E** *vigor* **É** *questionavel* **ENTÃO** *conclusao* **É** *inapto*
9	**SE** *anorma* **É** *ruim* **E** *cons* **É É** *DE* **E** *genotipo* **É** *nel* **ENTÃO** *conclusao* **É** *questionavel*
10	**SE** *anorma* **É** *ruim* **E** *vigor* **É** *muito_bom* **ENTÃO** *conclusao* **É** *questionavel*
11	**SE** *anorma* **É** *bom* **E** *motil* **É** *questionavel* **ENTÃO** *conclusao* **É** *questionavel*
12	**SE** *cons* **É É** *LF* **E** *genotipo* **É** *comp* **ENTÃO** *conclusao* **É** *questionavel*
13	**SE** *cons* **É É** *LF* **E** *norm* **É** *bom* **ENTÃO** *conclusao* **É** *questionavel*
14	**SE** *maior* **É** *muito_bom* **E** *motil* **É** *questionavel* **ENTÃO** *conclusao* **É** *questionavel*

Tabela 7.9: Saída de console da ferramenta para a análise de conflitos entre a BR definida pelos especialistas e a BR aprendida pelo AGMO.

Regra do especialista 1 é: Redundante e mais especifica que a regra dos dados 4: E -> SE PE É questionavel E idade É adulto ENTÃO conclusao É inapto (sup. 6.84%) D -> SE PE É questionavel ENTÃO conclusao É inapto (sup. 10.22%)
Regra do especialista 2 é: Especializada pela regra dos dados 8: E -> SE vigor É questionavel ENTÃO conclusao É questionavel (sup. 16.63%) D -> SE norm É ruim E vigor É questionavel ENTÃO conclusao É inapto (sup. 17.54%)
Regra do especialista 3 é: Uma especialização da regra dos dados 5: E -> SE anorma É ruim E idade É jovem ENTÃO conclusao É questionavel (sup. 9.96%) D -> SE anorma É ruim ENTÃO conclusao É inapto (sup. 23.44%)
Regra do especialista 4 é: Redundante e mais especifica que a regra dos dados 5: E -> SE anorma É ruim E idade É adulto ENTÃO conclusao É inapto (sup. 18.28%) D -> SE anorma É ruim ENTÃO conclusao É inapto (sup. 23.44%)
Regra do especialista 6 é: Especializada pela regra dos dados 7: E -> SE motil É questionavel ENTÃO conclusao É questionavel (sup. 12.13%) D -> SE idade É adulto E motil É questionavel ENTÃO conclusao É inapto (sup. 12.43%) Redundante e mais genérica que a regra dos dados 11: E -> SE motil É questionavel ENTÃO conclusao É questionavel (sup. 12.13%) D -> SE anorma É bom E motil É questionavel ENTÃO conclusao É questionavel (sup. 8.53%) Redundante e mais genérica que a regra dos dados 14: E -> SE motil É questionavel ENTÃO conclusao É questionavel (sup. 12.13%) D -> SE maior É muito_bom E motil É questionavel ENTÃO conclusao É questionavel (sup. 5.80%)
Regra do especialista 7 é: Redundante e mais especifica que a regra dos dados 4: E -> SE PE É questionavel E motil É questionavel E norm É ruim ENTÃO conclusao É inapto (sup. 3.94%) D -> SE PE É questionavel ENTÃO conclusao É inapto (sup. 10.22%)
Regra do especialista 11 é: Uma especialização da regra dos dados 5: E -> SE anorma É ruim E consis É F ENTÃO conclusao É questionavel (sup. 0.26%) D -> SE anorma É ruim ENTÃO conclusao É inapto (sup. 23.44%)
Regra do especialista 14 é: Especializada pela regra dos dados 11: E -> SE anorma É bom ENTÃO conclusao É apto (sup. 46.36%) D -> SE anorma É bom E motil É questionavel ENTÃO conclusao É questionavel (sup. 8.53%)
As regras do especialista 5, 8, 9, 10, 12 and 13 não foram encontradas na BR dos dados: E -> SE PE É bom E norm É ruim ENTÃO conclusao É questionavel (sup. 11.95%) E -> SE maior É ruim ENTÃO conclusao É questionavel (sup. 15.33%) E -> SE menor É ruim ENTÃO conclusao É questionavel (sup. 5.58%) E -> SE consis É F E maior É ruim E motil É questionavel ENTÃO conclusao É inapto (sup. 0.34%) E -> SE PE É bom ENTÃO conclusao É apto (sup. 38.69%) E -> SE motil É bom E norm É bom ENTÃO conclusao É apto (sup. 8.66%)

Tabela 7.10: Base de regras final, obtida através do processo de conciliação.

1	**SE** *PE* **É** *bom* **ENTÃO** *conclusao* **É** *apto*
2	**SE** *anorma* **É** *bom* **ENTÃO** *conclusao* **É** *apto*
3	**SE** *consis* **É** *DE* **ENTÃO** *conclusao* **É** *apto*
4	**SE** *idade* **É** *adulto* **E** *menor* **É** *muito_bom* **ENTÃO** *conclusao* **É** *apto*
5	**SE** *motil* **É** *bom* **E** *norm* **É** *bom* **ENTÃO** *conclusao* **É** *apto*
6	**SE** *norm* **É** *muito_bom* **ENTÃO** *conclusao* **É** *apto*
7	**SE** *PE* **É** *questionavel* **ENTÃO** *conclusao* **É** *inapto*
8	**SE** *anorma* **É** *ruim* **ENTÃO** *conclusao* **É** *inapto*
9	**SE** *consis* **É** *LF* **E** *idade* **É** *jovem* **ENTÃO** *conclusao* **É** *inapto*
10	**SE** *idade* **É** *adulto* **E** *motil* **É** *questionavel* **ENTÃO** *conclusao* **É** *inapto*
11	**SE** *norm* **É** *ruim* **E** *vigor* **É** *questionavel* **ENTÃO** *conclusao* **É** *inapto*
12	**SE** *PE* **É** *bom* **E** *norm* **É** *ruim* **ENTÃO** *conclusao* **É** *questionavel*
13	**SE** *anorma* **É** *ruim* **E** *consis* **É** *DE* **E** *genotipo* **É** *nel* **ENTÃO** *conclusao* **É** *questionavel*
14	**SE** *anorma* **É** *ruim* **E** *idade* **É** *jovem* **ENTÃO** *conclusao* **É** *questionavel*
15	**SE** *anorma* **É** *ruim* **E** *vigor* **É** *muito_bom* **ENTÃO** *conclusao* **É** *questionavel*
16	**SE** *anorma* **É** *bom* **E** *motil* **É** *questionavel* **ENTÃO** *conclusao* **É** *questionavel*
17	**SE** *consis* **É** *LF* **E** *genotipo* **É** *comp* **ENTÃO** *conclusao* **É** *questionavel*
18	**SE** *consis* **É** *LF* **E** *norm* **É** *bom* **ENTÃO** *conclusao* **É** *questionavel*
19	**SE** *maior* **É** *muito_bom* **E** *motil* **É** *questionavel* **ENTÃO** *conclusao* **É** *questionavel*

Tabela 7.11: Fronteira de Pareto resultante da execução da otimização genética da BD.

Q	#	Interpretabilidade	Erro na classificação	
	1	0.0567	0.2179	
	2	0.0494	0.2183	
	3	0.0467	0.2188	
Q1	4	0.0446	0.2192	
	5	0.0378	0.2196	
	6	0.0330	0.2201	Min
	7	0.0300	0.2205	
	8	0.0269	0.2209	
	9	0.0240	0.2214	
	10	0.0223	0.2218	
	11	0.0190	0.2222	
Q2	12	0.0158	0.2227	
	13	0.0156	0.2231	
	14	0.0146	0.2235	
	15	0.0145	0.2240	
	16	0.0125	0.2244	Med
	17	0.0122	0.2248	
	18	0.0089	0.2253	
	19	0.0088	0.2257	
Q3	20	0.0077	0.2261	
	21	0.0063	0.2274	
	22	0.0054	0.2279	
	23	0.0033	0.2283	Bal
	24	0.0030	0.2287	
	25	0.0011	0.2309	
	26	0.0008	0.2313	
	27	0.0006	0.2335	
Q4	28	0.0005	0.2374	
	29	0.0004	0.2396	
	30	0.0003	0.2426	
	31	0.0001	0.2456	
	32	0.0000	0.2478	

Med = Solução mediana, Min = Solução mínima, Bal = Solução balanceada.

Tabela 7.12: Resultado da validação cruzada estratificada de 10 partições para o processo de otimização genética da BD.

Posição na Fronteira	Erro médio na classificação
Solução original	24.7852%
Solução balanceada	23.1816%
Solução mínima	22.7054%
Solução mediana	23.0081%
Primeira solução	22.8351%
Média das soluções em Q1	22.7991%
Média das soluções em Q2	22.8434%
Média das soluções em Q3	23.1646%
Média das soluções em Q4	23.7506%

Tabela 7.13: Comparativo do erro médio da classificação para a abordagem proposta (média das soluções do primeiro quartil) e algoritmos consolidados na literatura, utilizando validação cruzada de 10 partições e o conjunto de dados do estudo de caso.

Algoritmo	Erro médio na classificação
Abordagem proposta (Q1)	22.79%
ZeroR	52.86%
NaiveBayes	28.53%
KNN k=1	22.80%
C4.5	23.98%
SVM	24.21%

Tabela 7.14: Comparativo entre a abordagem proposta, o método de construção automática e a construção manual de SFBR, quanto à complexidade de seus componente e ao erro médio na classificação.

Modelo	Base de Regras		Base de Dados		Erro	
	Regras	*Antecedentes*	*Variáveis*	*Conjuntos Fuzzy*		
Abordagem Proposta	19	33	11	38	22.79%	○
Construção Automática	23	67	15	71	22.07%	○
Construção Manual	124	368	6	21	51.90%	•

○ = Erro médio das soluções em Q1 na validação cruzada estratificada de 10 partições
• = aferido sobre os dados de treinamento

Capítulo 8

CONCLUSÃO

Este trabalho apresentou uma abordagem interativa para construção de SFBRs capaz de integrar o conhecimento extraído de especialistas de domínio de conhecimento e aquele induzido de dados. A abordagem mantém o especialista no controle das decisões nos níveis linguístico e semântico, enquanto o aprendizado e otimização dos componentes do SFBR fica a cargo de um AGMO, utilizando dados.

A abordagem é composta de seis etapas. Na primeira, vários métodos de seleção de atributos foram agregados para subsidiar a escolha do especialista sobre a relevância de cada atributo. Na segunda etapa, estatísticas descritivas, métodos para particionamento automático e aprendizado dos parâmetros desconhecidos, auxiliam o especialista na definição da BD. Na terceira etapa, as regras do especialista são definidas; estas direcionam o aprendizado genético da BR realizado na etapa seguinte. Na quinta etapa, as regras do especialista são usadas para garantir a semântica desejada através de um processo de conciliação com a BR aprendida dos dados. Na última etapa, um processo de otimização genética da BD é realizado.

Foi também apresentada uma ferramenta de software multiplataforma que implementa todas as etapas da abordagem proposta. Três interfaces foram criadas para a ferramenta. Uma via biblioteca de software, uma interface gráfica *desktop* e uma interface via linha de comando.

A eficiência da abordagem foi avaliada através de um estudo de caso, onde foi desenvolvido um sistema de suporte à decisão para avaliação da aptidão reprodutiva de touros nelore no Pantanal. O modelo obtido foi comparado à algoritmos clássicos para classificação e, também, à outros métodos para construção de SFBR, evidenciando que a abordagem proposta é capaz de alcançar a interpretabilidade de um modelo definido pelo especialista e a acurácia de um modelo construído via aprendizado de máquina.

Durante o desenvolvimento deste trabalho, alguns aspectos revelaram-se merecedores de investigação futura, tais como:

- Avaliar a inclusão à abordagem de uma etapa para auxiliar o tratamento de conjuntos de dados com classes desbalanceadas.
- Investigar os efeitos do uso de outras medidas de desempenho como função objetivo dos AGMO implementados, como a área sobre a curva ROC e a área sobre a curva *Precision-Recall*, no caso de problemas de classificação.
- Investigar métodos para simplificação das partições *fuzzy* que não resultem em perda da semântica. Isto seria interessante, visto que, frequentemente, boa parte dos CF definidos na BD não são utilizados por nenhuma regra na BR ao final do processo.
- Avaliar a utilização de outros AGMO, principalmente os que possuem como viés a obtenção de uma fronteira de Pareto com soluções mais bem distribuídas.

Apêndice A

CÓDIGOS FONTE DO ESTUDO DE CASO

Nas Tabelas a seguir serão listados os código fonte na linguagem FCL dos modelos desenvolvidos no estudo de caso apresentado no capítulo 7.

A.1 Código fonte do SFBR Final

```
FUNCTION_BLOCK androgenia_pantanal

VAR_INPUT
 PE : REAL;
 anorma : REAL;
 consis : REAL;
 genotipo : REAL;
 idade : REAL;
 maior : REAL;
 menor : REAL;
 motil : REAL;
 norm : REAL;
 vigor : REAL;
END_VAR

VAR_OUTPUT
 conclusao : REAL;
END_VAR

FUZZIFY PE
 TERM bom := TRIAN 28.148689638676277 33.0378199640617 38.748938080882944;
 TERM excelente := TRAPE 39.0 45.0 50.0 50.0;
 TERM muito_bom := TRIAN 33.0 39.0 45.0;
 TERM questionavel := TRAPE 22.0 22.0 27.0 33.0;
END_FUZZIFY

FUZZIFY anorma
 TERM bom := TRIAN 0.0 20.0 40.0;
 TERM muito_bom := TRIAN 0.0 0.0 20.0;
```

```
 TERM ruim := TRAPE 14.443659837813907 40.19292193879234 171.0 171.0;
END_FUZZIFY

FUZZIFY consis
 TERM D := 1;
 TERM DE := 2;
 TERM E := 3;
 TERM F := 4;
 TERM LF := 5;
END_FUZZIFY

FUZZIFY genotipo
 TERM comp := 1;
 TERM euro := 2;
 TERM nel := 3;
END_FUZZIFY

FUZZIFY idade
 TERM adulto := TRIAN 24.0 48.0 192.0;
 TERM jovem := TRIAN 12.007568407093034 23.47948486980044 45.953959801291774;
 TERM pre_pubere := TRIAN 12.0 12.0 24.0;
 TERM velho := TRAPE 48.0 192.0 200.0 200.0;
END_FUZZIFY

FUZZIFY maior
 TERM bom := TRIAN 12.5 17.5 22.5;
 TERM muito_bom := TRAPE 0.0 0.0 12.5 17.5;
 TERM ruim := TRAPE 17.5 22.5 134.0 134.0;
END_FUZZIFY

FUZZIFY menor
 TERM bom := TRIAN 2.5 17.5 32.5;
 TERM muito_bom := TRAPE 0.0 0.0 2.5 17.5;
 TERM ruim := TRAPE 17.5 32.5 100.0 100.0;
END_FUZZIFY

FUZZIFY motil
 TERM bom := TRIAN 45.0 55.0 65.0;
 TERM excelente := TRAPE 65.0 85.0 100.0 100.0;
 TERM muito_bom := TRIAN 55.0 65.0 85.0;
 TERM questionavel := TRAPE 0.0 0.0 46.81284365678238 58.33164477668622;
END_FUZZIFY

FUZZIFY norm
 TERM bom := TRIAN 65.44603659758243 75.70871038644333 86.01998611580645;
 TERM muito_bom := TRAPE 78.25475311035424 88.20417391490635 100.0 100.0;
 TERM ruim := TRAPE 0.0 0.0 69.94285146193891 79.38833891969594;
END_FUZZIFY

FUZZIFY vigor
 TERM bom := TRIAN 2.5 3.5 4.5;
 TERM muito_bom := TRAPE 3.2537734462202277 4.283475342691474 5.0 5.0;
 TERM questionavel := TRAPE 0.0 0.0 2.5 3.5;
END_FUZZIFY
```

```
DEFUZZIFY conclusao
 TERM apto :=  1.0;
 TERM inapto :=  2.0;
 TERM questionavel :=  3.0;
 METHOD : COGS;
 DEFAULT := NC;
 RANGE := (1.0 .. 3.0);
END_DEFUZZIFY

RULEBLOCK No1
 ACT : MIN;
 ACCU : MAX;
 AND : MIN;
 RULE 001 : IF PE IS bom THEN conclusao IS apto;
 RULE 002 : IF anorma IS bom THEN conclusao IS apto;
 RULE 003 : IF consis IS DE THEN conclusao IS apto;
 RULE 004 : IF (idade IS adulto) AND (menor IS muito_bom) THEN conclusao IS apto;
 RULE 005 : IF (motil IS bom) AND (norm IS bom) THEN conclusao IS apto;
 RULE 006 : IF norm IS muito_bom THEN conclusao IS apto;
 RULE 007 : IF PE IS questionavel THEN conclusao IS inapto;
 RULE 008 : IF anorma IS ruim THEN conclusao IS inapto;
 RULE 009 : IF (consis IS LF) AND (idade IS jovem) THEN conclusao IS inapto;
 RULE 010 : IF (idade IS adulto) AND (motil IS questionavel) THEN conclusao IS inapto;
 RULE 011 : IF (norm IS ruim) AND (vigor IS questionavel) THEN conclusao IS inapto;
 RULE 012 : IF (PE IS bom) AND (norm IS ruim) THEN conclusao IS questionavel;
 RULE 013 : IF ((anorma IS ruim) AND (consis IS DE)) AND (genotipo IS nel) THEN conclusao IS questionavel;
 RULE 014 : IF (anorma IS ruim) AND (idade IS jovem) THEN conclusao IS questionavel;
 RULE 015 : IF (anorma IS ruim) AND (vigor IS muito_bom) THEN conclusao IS questionavel;
 RULE 016 : IF (anorma IS bom) AND (motil IS questionavel) THEN conclusao IS questionavel;
 RULE 017 : IF (consis IS LF) AND (genotipo IS comp) THEN conclusao IS questionavel;
 RULE 018 : IF (consis IS LF) AND (norm IS bom) THEN conclusao IS questionavel;
 RULE 019 : IF (maior IS muito_bom) AND (motil IS questionavel) THEN conclusao IS questionavel;
END_RULEBLOCK

END_FUNCTION_BLOCK
```

A.2 Código fonte do SFBR construído automaticamente

```
FUNCTION_BLOCK androgenia_pantanal_auto

VAR_INPUT
 GCP : REAL;
 acros : REAL;
 anorma : REAL;
 cauda : REAL;
 consis : REAL;
 isolnor : REAL;
 maior : REAL;
 motil : REAL;
 norm : REAL;
 pi : REAL;
 raca : REAL;
```

```
 regiao : REAL;
 turbil : REAL;
 vol : REAL;
END_VAR

VAR_OUTPUT
 conclusao : REAL;
END_VAR

FUZZIFY GCP
 TERM High := TRIAN 1.5957538994800693 61.0 61.0;
 TERM Low := TRIAN 0.0 0.0 1.5957538994800693;
 TERM Medium := TRIAN 0.0 1.5957538994800693 61.0;
END_FUZZIFY


FUZZIFY acros
 TERM High := TRIAN 18.675925925925927 38.793103448275865 99.0;
 TERM Low := TRIAN 0.0 0.20813559322033898 2.9262948207171315;
 TERM Medium := TRIAN 2.9262948207171315 8.407216494845361 18.675925925925927;
 TERM Medium_High := TRIAN 8.407216494845361 18.675925925925927 38.793103448275865;
 TERM Medium_Low := TRIAN 0.20813559322033898 2.9262948207171315 8.407216494845361;
 TERM Very_High := TRIAN 38.793103448275865 99.0 99.0;
 TERM Very_Low := TRIAN 0.0 0.0 0.20813559322033898;
END_FUZZIFY

FUZZIFY anorma
 TERM High := TRIAN 30.0 50.0 171.0;
 TERM Low := TRIAN 0.0 10.5 17.5;
 TERM Medium_High := TRIAN 17.5 30.0 50.0;
 TERM Medium_Low := TRIAN 10.5 17.5 30.0;
 TERM Very_High := TRIAN 50.0 171.0 171.0;
 TERM Very_Low := TRIAN 0.0 0.0 10.5;
END_FUZZIFY

FUZZIFY cauda
 TERM High := TRIAN 21.58139534883721 40.14754098360656 83.0;
 TERM Low := TRIAN 0.0 1.2948051948051948 9.304878048780488;
 TERM Medium_High := TRIAN 9.304878048780488 21.58139534883721 40.14754098360656;
 TERM Medium_Low := TRIAN 1.2948051948051948 9.304878048780488 21.58139534883721;
 TERM Very_High := TRIAN 40.14754098360656 83.0 83.0;
 TERM Very_Low := TRIAN 0.0 0.0 1.2948051948051948;
END_FUZZIFY

FUZZIFY consis
 TERM D := 1.0;
 TERM DE := 2.0;
 TERM E := 3.0;
 TERM F := 4.0;
 TERM LF := 5.0;
END_FUZZIFY

FUZZIFY isolnor
 TERM High := TRIAN 17.94701986754967 51.0 51.0;
```

```
 TERM Low := TRIAN 0.0 0.0 1.4789058878071395;
 TERM Medium_High := TRIAN 1.4789058878071395 17.94701986754967 51.0;
 TERM Medium_Low := TRIAN 0.0 1.4789058878071395 17.94701986754967;
END_FUZZIFY

FUZZIFY maior
 TERM High := TRIAN 89.333333333333 111.666666666666 134.0;
 TERM Low := TRIAN 0.0 22.333333333333 44.666666666666;
 TERM Medium := TRIAN 44.666666666666 67.0 89.333333333333;
 TERM Medium_High := TRIAN 67.0 89.333333333333 111.666666666666;
 TERM Medium_Low := TRIAN 22.333333333333 44.666666666666 67.0;
 TERM Very_High := TRIAN 111.666666666666 134.0 134.0;
 TERM Very_Low := TRIAN 0.0 0.0 22.333333333333;
END_FUZZIFY


FUZZIFY motil
 TERM High := TRIAN 60.0 90.0 90.0;
 TERM Low := TRIAN 0.0 0.0 30.0;
 TERM Medium_High := TRIAN 30.0 60.0 90.0;
 TERM Medium_Low := TRIAN 0.0 30.0 60.0;
END_FUZZIFY

FUZZIFY norm
 TERM High := TRIAN 73.5553772070626 89.34434434434435 100.0;
 TERM Low := TRIAN 0.0 17.486631016042782 51.054108216432866;
 TERM Medium_High := TRIAN 51.054108216432866 73.5553772070626 89.34434434434435;
 TERM Medium_Low := TRIAN 17.486631016042782 51.054108216432866 73.5553772070626;
 TERM Very_High := TRIAN 89.34434434434435 100.0 100.0;
 TERM Very_Low := TRIAN 0.0 0.0 17.486631016042782;
END_FUZZIFY

FUZZIFY pi
 TERM High := TRIAN 19.632 82.0 82.0;
 TERM Low := TRIAN 0.0 0.0 1.4703595724003888;
 TERM Medium_High := TRIAN 1.4703595724003888 19.632 82.0;
 TERM Medium_Low := TRIAN 0.0 1.4703595724003888 19.632;
END_FUZZIFY

FUZZIFY raca
 TERM MTN := 1.0;
 TERM brangus := 2.0;
 TERM canchim := 3.0;
 TERM nelore := 4.0;
 TERM simental := 5.0;
END_FUZZIFY

FUZZIFY regiao
 TERM pantanal := 1.0;
 TERM plan := 2.0;
END_FUZZIFY

FUZZIFY turbil
 TERM High := TRIAN 3.333333333333 5.0 5.0;
```

```
 TERM Low := TRIAN 0.0 0.0 1.666666666666;
 TERM Medium_High := TRIAN 1.666666666666 3.333333333333 5.0;
 TERM Medium_Low := TRIAN 0.0 1.666666666666 3.333333333333;
END_FUZZIFY

FUZZIFY vol
 TERM High := TRIAN 14.875 19.5 19.5;
 TERM Low := TRIAN 1.0 1.0 5.625;
 TERM Medium := TRIAN 5.625 10.25 14.875;
 TERM Medium_High := TRIAN 10.25 14.875 19.5;
 TERM Medium_Low := TRIAN 1.0 5.625 10.25;
END_FUZZIFY

DEFUZZIFY conclusao
 TERM apto :=  1.0;
 TERM inapto :=  2.0;
 TERM questionavel :=  3.0;
 METHOD : COGS;
 DEFAULT := NC;
 RANGE := (1.0 .. 3.0);
END_DEFUZZIFY

RULEBLOCK No1
 ACT : MIN;
 ACCU : MAX;
 RULE 001 : IF GCP IS Medium AND acros IS Low AND anorma IS Medium_Low AND isolnor IS
              Medium_Low THEN conclusao IS apto;
 RULE 002 : IF GCP IS Low AND cauda IS Very_Low AND isolnor IS Low AND maior IS Low AND motil
              IS Medium_High THEN conclusao IS apto;
 RULE 003 : IF anorma IS Medium_High AND raca IS brangus THEN conclusao IS apto;
 RULE 004 : IF cauda IS Medium_Low AND maior IS Low AND motil IS Medium_High AND turbil IS
              Medium_High AND vol IS Medium_Low THEN conclusao IS apto;
 RULE 005 : IF maior IS Low AND motil IS Medium_High AND norm IS Medium_High AND regiao IS
              pantanal THEN conclusao IS apto;
 RULE 006 : IF motil IS Medium_High AND norm IS Very_High AND pi IS Low THEN conclusao
              IS apto;
 RULE 007 : IF motil IS High AND norm IS Medium_High THEN conclusao IS apto;
 RULE 008 : IF norm IS High THEN conclusao IS apto;
 RULE 009 : IF acros IS High THEN conclusao IS inapto;
 RULE 010 : IF anorma IS High AND isolnor IS Medium_High AND raca IS nelore AND vol IS
              Medium_Low THEN conclusao IS inapto;
 RULE 011 : IF consis IS LF AND isolnor IS Medium_Low AND regiao IS pantanal THEN conclusao
              IS inapto;
 RULE 012 : IF consis IS LF AND maior IS Low AND pi IS Low THEN conclusao IS inapto;
 RULE 013 : IF consis IS LF AND regiao IS pantanal AND vol IS Medium_Low THEN conclusao
              IS inapto;
 RULE 014 : IF consis IS E AND norm IS Medium_Low THEN conclusao IS inapto;
 RULE 015 : IF maior IS Medium_Low THEN conclusao IS inapto;
 RULE 016 : IF maior IS Medium_High THEN conclusao IS inapto;
 RULE 017 : IF GCP IS Medium AND acros IS Medium_Low AND pi IS Medium_Low AND turbil IS
              Medium_High AND vol IS Medium THEN conclusao IS questionavel;
 RULE 018 : IF acros IS Medium AND pi IS Medium_Low AND regiao IS plan AND turbil IS
              Medium_High THEN conclusao IS questionavel;
```

```
 RULE 019 : IF anorma IS Medium_High AND norm IS Medium_High AND regiao IS plan THEN conclusao
            IS questionavel;
 RULE 020 : IF anorma IS High AND consis IS DE AND regiao IS pantanal THEN conclusao
              IS questionavel;
 RULE 021 : IF consis IS DE AND norm IS Medium_Low AND turbil IS High THEN conclusao
              IS questionavel;
 RULE 022 : IF motil IS Medium_Low AND norm IS High THEN conclusao IS questionavel;
 RULE 023 : IF norm IS Medium_Low AND regiao IS plan AND turbil IS Medium_High THEN conclusao
              IS questionavel;
END_RULEBLOCK

END_FUNCTION_BLOCK
```

A.3 Código fonte do SFBR Final construído manualmente

```
FUNCTION_BLOCK androgenia_pantanal_manual

VAR_INPUT
 PE : REAL;
 anorma : REAL;
 idade : REAL;
 motil : REAL;
 vigor : REAL;
END_VAR

VAR_OUTPUT
 conclusao : REAL;
END_VAR

FUZZIFY PE
 TERM bom := TRIAN 27.0 33.0 39.0;
 TERM excelente := TRAPE 39.0 45.0 50.0 50.0;
 TERM muito_bom := TRIAN 33.0 39.0 45.0;
 TERM questionavel := TRAPE 22.0 22.0 27.0 33.0;
END_FUZZIFY

FUZZIFY anorma
 TERM bom := TRIAN 0.0 20.0 40.0;
 TERM muito_bom := TRIAN 0.0 0.0 20.0;
 TERM ruim := TRAPE 20.0 40.0 171.0 171.0;
END_FUZZIFY

FUZZIFY idade
 TERM adulto := TRIAN 24.0 48.0 192.0;
 TERM jovem := TRIAN 12.0 24.0 48.0;
 TERM pre_pubere := TRIAN 12.0 12.0 24.0;
 TERM velho := TRAPE 48.0 192.0 200.0 200.0;
END_FUZZIFY

FUZZIFY motil
 TERM bom := TRIAN 45.0 55.0 65.0;
 TERM excelente := TRAPE 65.0 85.0 100.0 100.0;
 TERM muito_bom := TRIAN 55.0 65.0 85.0;
```

```
 TERM questionavel := TRAPE 0.0 0.0 45.0 55.0;
END_FUZZIFY

FUZZIFY vigor
 TERM bom := TRIAN 2.5 3.5 4.5;
 TERM muito_bom := TRAPE 3.5 4.5 5.0 5.0;
 TERM questionavel := TRAPE 0.0 0.0 2.5 3.5;
END_FUZZIFY

DEFUZZIFY conclusao
 TERM apto :=  1.0;
 TERM inapto :=  2.0;
 TERM questionavel :=  3.0;
 METHOD : COGS;
 DEFAULT := NC;
 RANGE := (1.0 .. 3.0);
END_DEFUZZIFY

RULEBLOCK No1
 ACT : MIN;
 ACCU : MAX;
 AND : MIN;
 RULE 001 : IF PE IS questionavel AND idade IS pre_pubere THEN conclusao IS questionavel;
 RULE 002 : IF PE IS questionavel AND idade IS jovem THEN conclusao IS inapto;
 RULE 003 : IF PE IS questionavel AND idade IS adulto THEN conclusao IS inapto;
 RULE 004 : IF PE IS questionavel AND idade IS velho THEN conclusao IS inapto;
 RULE 005 : IF PE IS bom AND idade IS pre_pubere AND motil IS questionavel THEN conclusao
              IS questionavel;
 RULE 006 : IF PE IS bom AND idade IS pre_pubere AND motil IS bom THEN conclusao IS apto;
 RULE 007 : IF PE IS bom AND idade IS pre_pubere AND motil IS muito_bom THEN conclusao IS apto;
 RULE 008 : IF PE IS bom AND idade IS pre_pubere AND motil IS excelente THEN conclusao IS apto;
 RULE 009 : IF PE IS muito_bom AND idade IS pre_pubere AND motil IS questionavel THEN
              conclusao IS questionavel;
 RULE 010 : IF PE IS muito_bom AND idade IS pre_pubere AND motil IS bom THEN conclusao IS apto;
 RULE 011 : IF PE IS muito_bom AND idade IS pre_pubere AND motil IS muito_bom THEN conclusao
              IS apto;
 RULE 012 : IF PE IS muito_bom AND idade IS pre_pubere AND motil IS excelente THEN conclusao
              IS apto;
 RULE 013 : IF PE IS excelente AND idade IS pre_pubere AND motil IS questionavel THEN
              conclusao IS questionavel;
 RULE 014 : IF PE IS excelente AND idade IS pre_pubere AND motil IS bom THEN conclusao IS apto;
 RULE 015 : IF PE IS excelente AND idade IS pre_pubere AND motil IS muito_bom THEN conclusao
               IS apto;
 RULE 016 : IF PE IS excelente AND idade IS pre_pubere AND motil IS excelente THEN conclusao
              IS apto;
 RULE 017 : IF PE IS bom AND idade IS jovem AND motil IS questionavel THEN conclusao
              IS questionavel;
 RULE 018 : IF PE IS bom AND idade IS jovem AND motil IS bom THEN conclusao IS apto;
 RULE 019 : IF PE IS bom AND idade IS jovem AND motil IS muito_bom THEN conclusao IS apto;
 RULE 020 : IF PE IS bom AND idade IS jovem AND motil IS excelente THEN conclusao IS apto;
 RULE 021 : IF PE IS muito_bom AND idade IS jovem AND motil IS questionavel THEN conclusao
              IS questionavel;
 RULE 022 : IF PE IS muito_bom AND idade IS jovem AND motil IS bom THEN conclusao IS apto;
 RULE 023 : IF PE IS muito_bom AND idade IS jovem AND motil IS muito_bom THEN conclusao
```

```
            IS apto;
RULE 024 : IF PE IS muito_bom AND idade IS jovem AND motil IS excelente THEN conclusao
            IS apto;
RULE 025 : IF PE IS excelente AND idade IS jovem AND motil IS questionavel THEN conclusao
            IS questionavel;
RULE 026 : IF PE IS excelente AND idade IS jovem AND motil IS bom THEN conclusao IS apto;
RULE 027 : IF PE IS excelente AND idade IS jovem AND motil IS muito_bom THEN conclusao
            IS apto;
RULE 028 : IF PE IS excelente AND idade IS jovem AND motil IS excelente THEN conclusao
            IS apto;
RULE 029 : IF PE IS bom AND idade IS adulto AND motil IS questionavel THEN conclusao
            IS questionavel;
RULE 030 : IF PE IS bom AND idade IS adulto AND motil IS bom THEN conclusao IS apto;
RULE 031 : IF PE IS bom AND idade IS adulto AND motil IS muito_bom THEN conclusao IS apto;
RULE 032 : IF PE IS bom AND idade IS adulto AND motil IS excelente THEN conclusao IS apto;
RULE 033 : IF PE IS muito_bom AND idade IS adulto AND motil IS questionavel THEN conclusao
            IS questionavel;
RULE 034 : IF PE IS muito_bom AND idade IS adulto AND motil IS bom THEN conclusao IS apto;
RULE 035 : IF PE IS muito_bom AND idade IS adulto AND motil IS muito_bom THEN conclusao
            IS apto;
RULE 036 : IF PE IS muito_bom AND idade IS adulto AND motil IS excelente THEN conclusao
            IS apto;
RULE 037 : IF PE IS excelente AND idade IS adulto AND motil IS questionavel THEN conclusao
            IS questionavel;
RULE 038 : IF PE IS excelente AND idade IS adulto AND motil IS bom THEN conclusao IS apto;
RULE 039 : IF PE IS excelente AND idade IS adulto AND motil IS muito_bom THEN conclusao
            IS apto;
RULE 040 : IF PE IS excelente AND idade IS adulto AND motil IS excelente THEN conclusao
            IS apto;
RULE 041 : IF PE IS bom AND idade IS velho AND motil IS questionavel THEN conclusao IS inapto;
RULE 042 : IF PE IS bom AND idade IS velho AND motil IS bom THEN conclusao IS apto;
RULE 043 : IF PE IS bom AND idade IS velho AND motil IS muito_bom THEN conclusao IS apto;
RULE 044 : IF PE IS bom AND idade IS velho AND motil IS excelente THEN conclusao IS apto;
RULE 045 : IF PE IS muito_bom AND idade IS velho AND motil IS questionavel THEN conclusao
            IS inapto;
RULE 046 : IF PE IS muito_bom AND idade IS velho AND motil IS bom THEN conclusao IS apto;
RULE 047 : IF PE IS muito_bom AND idade IS velho AND motil IS muito_bom THEN conclusao
             IS apto;
RULE 048 : IF PE IS muito_bom AND idade IS velho AND motil IS excelente THEN conclusao
            IS apto;
RULE 049 : IF PE IS excelente AND idade IS velho AND motil IS questionavel THEN conclusao IS
            inapto;
RULE 050 : IF PE IS excelente AND idade IS velho AND motil IS bom THEN conclusao IS apto;
RULE 051 : IF PE IS excelente AND idade IS velho AND motil IS muito_bom THEN conclusao
            IS apto;
RULE 052 : IF PE IS excelente AND idade IS velho AND motil IS excelente THEN conclusao
            IS apto;
RULE 053 : IF PE IS bom AND idade IS pre_pubere AND vigor IS questionavel THEN conclusao
            IS questionavel;
RULE 054 : IF PE IS bom AND idade IS pre_pubere AND vigor IS bom THEN conclusao IS apto;
RULE 055 : IF PE IS bom AND idade IS pre_pubere AND vigor IS muito_bom THEN conclusao IS apto;
RULE 056 : IF PE IS muito_bom AND idade IS pre_pubere AND vigor IS questionavel THEN conclusao
            IS questionavel;
RULE 057 : IF PE IS muito_bom AND idade IS pre_pubere AND vigor IS bom THEN conclusao IS apto;
```

```
RULE 058 : IF PE IS muito_bom AND idade IS pre_pubere AND vigor IS muito_bom THEN conclusao
             IS apto;
RULE 059 : IF PE IS excelente AND idade IS pre_pubere AND vigor IS questionavel THEN conclusao
             IS questionavel;
RULE 060 : IF PE IS excelente AND idade IS pre_pubere AND vigor IS bom THEN conclusao IS apto;
RULE 061 : IF PE IS excelente AND idade IS pre_pubere AND vigor IS muito_bom THEN conclusao
             IS apto;
RULE 062 : IF PE IS bom AND idade IS jovem AND vigor IS questionavel THEN conclusao
             IS questionavel;
RULE 063 : IF PE IS bom AND idade IS jovem AND vigor IS bom THEN conclusao IS apto;
RULE 064 : IF PE IS bom AND idade IS jovem AND vigor IS muito_bom THEN conclusao IS apto;
RULE 065 : IF PE IS muito_bom AND idade IS jovem AND vigor IS questionavel THEN conclusao
             IS questionavel;
RULE 066 : IF PE IS muito_bom AND idade IS jovem AND vigor IS bom THEN conclusao IS apto;
RULE 067 : IF PE IS muito_bom AND idade IS jovem AND vigor IS muito_bom THEN conclusao
             IS apto;
RULE 068 : IF PE IS excelente AND idade IS jovem AND vigor IS questionavel THEN conclusao
             IS questionavel;
RULE 069 : IF PE IS excelente AND idade IS jovem AND vigor IS bom THEN conclusao IS apto;
RULE 070 : IF PE IS excelente AND idade IS jovem AND vigor IS muito_bom THEN conclusao
             IS apto;
RULE 071 : IF PE IS bom AND idade IS adulto AND vigor IS questionavel THEN conclusao
             IS questionavel;
RULE 072 : IF PE IS bom AND idade IS adulto AND vigor IS bom THEN conclusao IS apto;
RULE 073 : IF PE IS bom AND idade IS adulto AND vigor IS muito_bom THEN conclusao IS apto;
RULE 074 : IF PE IS muito_bom AND idade IS adulto AND vigor IS questionavel THEN conclusao
             IS questionavel;
RULE 075 : IF PE IS muito_bom AND idade IS adulto AND vigor IS bom THEN conclusao IS apto;
RULE 076 : IF PE IS muito_bom AND idade IS adulto AND vigor IS muito_bom THEN conclusao
             IS apto;
RULE 077 : IF PE IS excelente AND idade IS adulto AND vigor IS questionavel THEN conclusao
             IS questionavel;
RULE 078 : IF PE IS excelente AND idade IS adulto AND vigor IS bom THEN conclusao IS apto;
RULE 079 : IF PE IS excelente AND idade IS adulto AND vigor IS muito_bom THEN conclusao
             IS apto;
RULE 080 : IF PE IS bom AND idade IS velho AND vigor IS questionavel THEN conclusao IS inapto;
RULE 081 : IF PE IS bom AND idade IS velho AND vigor IS bom THEN conclusao IS apto;
RULE 082 : IF PE IS bom AND idade IS velho AND vigor IS muito_bom THEN conclusao IS apto;
RULE 083 : IF PE IS muito_bom AND idade IS velho AND vigor IS questionavel THEN conclusao
             IS inapto;
RULE 084 : IF PE IS muito_bom AND idade IS velho AND vigor IS bom THEN conclusao IS apto;
RULE 085 : IF PE IS muito_bom AND idade IS velho AND vigor IS muito_bom THEN conclusao
             IS apto;
RULE 086 : IF PE IS excelente AND idade IS velho AND vigor IS questionavel THEN conclusao
             IS inapto;
RULE 087 : IF PE IS excelente AND idade IS velho AND vigor IS bom THEN conclusao IS apto;
RULE 088 : IF PE IS excelente AND idade IS velho AND vigor IS muito_bom THEN conclusao
             IS apto;
RULE 089 : IF PE IS bom AND idade IS pre_pubere AND anorma IS ruim THEN conclusao IS inapto;
RULE 090 : IF PE IS bom AND idade IS pre_pubere AND anorma IS bom THEN conclusao IS apto;
RULE 091 : IF PE IS bom AND idade IS pre_pubere AND anorma IS muito_bom THEN conclusao
             IS apto;
RULE 092 : IF PE IS muito_bom AND idade IS pre_pubere AND anorma IS bom THEN conclusao
             IS apto;
```

```
RULE 093 : IF PE IS muito_bom AND idade IS pre_pubere AND anorma IS muito_bom THEN conclusao
           IS apto;
RULE 094 : IF PE IS muito_bom AND idade IS pre_pubere AND anorma IS ruim THEN conclusao
           IS questionavel;
RULE 095 : IF PE IS questionavel AND idade IS pre_pubere AND anorma IS bom THEN conclusao
           IS apto;
RULE 096 : IF PE IS questionavel AND idade IS pre_pubere AND anorma IS muito_bom THEN
           conclusao IS apto;
RULE 097 : IF PE IS questionavel AND idade IS pre_pubere AND anorma IS ruim THEN conclusao
             IS questionavel;
RULE 098 : IF PE IS bom AND idade IS jovem AND anorma IS bom THEN conclusao IS apto;
RULE 099 : IF PE IS bom AND idade IS jovem AND anorma IS muito_bom THEN conclusao IS apto;
RULE 100 : IF PE IS bom AND idade IS jovem AND anorma IS ruim THEN conclusao IS questionavel;
RULE 101 : IF PE IS muito_bom AND idade IS jovem AND anorma IS bom THEN conclusao IS apto;
RULE 102 : IF PE IS muito_bom AND idade IS jovem AND anorma IS muito_bom THEN conclusao
           IS apto;
RULE 103 : IF PE IS muito_bom AND idade IS jovem AND anorma IS ruim THEN conclusao
           IS questionavel;
RULE 104 : IF PE IS excelente AND idade IS jovem AND anorma IS bom THEN conclusao IS apto;
RULE 105 : IF PE IS excelente AND idade IS jovem AND anorma IS muito_bom THEN conclusao
           IS apto;
RULE 106 : IF PE IS excelente AND idade IS jovem AND anorma IS ruim THEN conclusao
           IS questionavel;
RULE 107 : IF PE IS bom AND idade IS adulto AND anorma IS bom THEN conclusao IS apto;
RULE 108 : IF PE IS bom AND idade IS adulto AND anorma IS muito_bom THEN conclusao IS apto;
RULE 109 : IF PE IS bom AND idade IS adulto AND anorma IS ruim THEN conclusao IS inapto;
RULE 110 : IF PE IS muito_bom AND idade IS adulto AND anorma IS bom THEN conclusao IS apto;
RULE 111 : IF PE IS muito_bom AND idade IS adulto AND anorma IS muito_bom THEN conclusao
           IS apto;
RULE 112 : IF PE IS muito_bom AND idade IS adulto AND anorma IS ruim THEN conclusao IS inapto;
RULE 113 : IF PE IS excelente AND idade IS adulto AND anorma IS bom THEN conclusao IS apto;
RULE 114 : IF PE IS excelente AND idade IS adulto AND anorma IS muito_bom THEN conclusao
           IS apto;
RULE 115 : IF PE IS excelente AND idade IS adulto AND anorma IS ruim THEN conclusao IS inapto;
RULE 116 : IF PE IS bom AND idade IS velho AND anorma IS bom THEN conclusao IS apto;
RULE 117 : IF PE IS bom AND idade IS velho AND anorma IS muito_bom THEN conclusao IS apto;
RULE 118 : IF PE IS bom AND idade IS velho AND anorma IS ruim THEN conclusao IS inapto;
RULE 119 : IF PE IS muito_bom AND idade IS velho AND anorma IS bom THEN conclusao IS apto;
RULE 120 : IF PE IS muito_bom AND idade IS velho AND anorma IS muito_bom THEN conclusao
           IS apto;
RULE 121 : IF PE IS muito_bom AND idade IS velho AND anorma IS ruim THEN conclusao IS inapto;
RULE 122 : IF PE IS excelente AND idade IS velho AND anorma IS bom THEN conclusao IS apto;
RULE 123 : IF PE IS excelente AND idade IS velho AND anorma IS muito_bom THEN conclusao
           IS apto;
RULE 124 : IF PE IS excelente AND idade IS velho AND anorma IS ruim THEN conclusao IS inapto;
END_RULEBLOCK

END_FUNCTION_BLOCK
```

REFERÊNCIAS BIBLIOGRÁFICAS

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.

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, n. 5, p. 1106–1122, out. 2009. ISSN 1063-6706.

ALCALÁ, R. et al. Multiobjective genetic fuzzy rule selection of single granularity-based fuzzy classification rules and its interaction with the lateral tuning of membership functions. *Soft Computing*, v. 15, n. 12, p. 2303–2318, nov. 2010. ISSN 1432-7643.

ALONSO, J.; MAGDALENA, L. Generating understandable and accurate fuzzy rule-based systems in a java environment. *Lecture Notes in Artificial Intelligence - 9th International Workshop on Fuzzy Logic and Applications*, LNAI6857, n. January, p. 212–219, 2011.

ALONSO, J. M.; MAGDALENA, L. HILK++: an interpretability-guided fuzzy modeling methodology for learning readable and comprehensible fuzzy rule-based classifiers. *Soft Computing*, v. 15, n. 10, p. 1959–1980, jun. 2010. ISSN 1432-7643.

ALONSO, J. M. et al. Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures. *Information Sciences*, v. 181, n. 20, p. 4340–4360, 2011.

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, 2009.

ALONSO, J. M.; MAGDALENA, L.; GUILLAUME, S. HILK: A new methodology for designing highly interpretable linguistic knowledge bases using the fuzzy logic formalism. *International Journal of Intelligent Systems*, v. 23, n. 7, p. 761–794, jul. 2008. ISSN 08848173.

ANTONELLI, M. et al. Learning knowledge bases of multi-objective evolutionary fuzzy systems by simultaneously optimizing accuracy, complexity and partition integrity. *Soft Computing*, v. 15, n. 12, p. 2335–2354, nov. 2010. ISSN 1432-7643.

ANTONELLI, M. et al. Multi-objective evolutionary generation of Mamdani fuzzy rule-based systems based on rule and condition selection. In: *2011 IEEE 5th International Workshop on Genetic and Evolutionary Fuzzy Systems (GEFS)*. [S.l.]: IEEE, 2011. p. 47–53. ISBN 978-1-61284-049-9.

ASBIA. *Relatório do INDEX ASBIA - Mercado de sêmen 2014*. Uberaba, 2014. 30 p. Disponível em: <http://www.asbia.org.br/novo/upload/mercado/index2014.pdf>.

BARBOSA, R. T.; MACHADO, R.; BERGAMASCHI, M. A. C. M. *A importância do exame andrológico em bovinos*. São Carlos, 2005. 13 p.

BATISTA, G. E. A. P. A.; PRATI, R. C.; MONARD, M. C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, ACM, v. 6, n. 1, p. 20, jun. 2004. ISSN 19310145.

CAMARGO, H. A. Multiobjective genetic generation of fuzzy classifiers using the iterative rule learning. In: *2012 IEEE International Conference on Fuzzy Systems*. [S.l.]: IEEE, 2012. p. 1–8. ISBN 978-1-4673-1506-7. ISSN 1098-7584.

CANNONE, R.; ALONSO, J. M.; MAGDALENA, L. Multi-objective design of highly interpretable fuzzy rule-based classifiers with semantic cointension. In: *2011 IEEE 5th International Workshop on Genetic and Evolutionary Fuzzy Systems (GEFS)*. [S.l.]: Ieee, 2011. p. 1–8. ISBN 978-1-61284-049-9.

CASILLAS, J. et al. Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction. *IEEE Transactions on Fuzzy Systems*, v. 13, n. 1, p. 13–29, fev. 2005. ISSN 1063-6706.

CHI, Z.; YAN, H.; PHAM, T. *Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition*. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 1996. ISBN 9810226977.

CINGOLANI, P.; ALCALA-FDEZ, J. jFuzzyLogic: a robust and flexible Fuzzy-Logic inference system language implementation. In: *2012 IEEE International Conference on Fuzzy Systems*. [S.l.]: IEEE, 2012. p. 1–8. ISBN 978-1-4673-1506-7. ISSN 1098-7584.

CINGOLANI, P.; ALCALÁ-FDEZ, J. jFuzzyLogic: a java library to design fuzzy logic controllers according to the standard for fuzzy control programming. *International Journal of Computational Intelligence Systems*, Taylor & Francis, v. 6, n. sup1, p. 61–75, 2013.

CORDÓN, O. A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems. *International Journal of Approximate Reasoning*, v. 52, n. 6, p. 894–913, set. 2011. ISSN 0888613X.

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. ISSN 01650114.

CORDÓN, O.; HERRERA, F.; VILLAR, P. Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base. *IEEE Transactions on Fuzzy Systems*, v. 9, n. 4, p. 667–674, 2001. ISSN 10636706.

COULON-LEROY, C. et al. Imperfect knowledge and data-based approach to model a complex agronomic feature – Application to vine vigor. *Computers and Electronics in Agriculture*, v. 99, p. 135–145, nov. 2013. ISSN 01681699.

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. ISSN 1089778X.

DIAS, J. C. et al. CLASSIFICAÇÃO ANDROLóGICA POR PONTOS (CAP) DE TOUROS NELORE (Bos taurus indicus) DE DOIS E TRÊS ANOS DE IDADE, CRIADOS SOB PASTEJO. *Ciência Animal Brasileira*, v. 10, n. 4, 2009. ISSN 1809-6891.

DURILLO, J. J.; NEBRO, A. J.; ALBA, E. The jMetal Framework for Multi-Objective Optimization: Design and Architecture. In: *CEC 2010*. Barcelona, Spain: [s.n.], 2010. p. 4138–4325.

EBERHART, R. C.; SHI, Y. *Computational Intelligence: Concepts to Implementations*. [S.l.]: Elsevier, 2007. 496 p. ISBN 0080553834.

FAZZOLARI, M. et al. A Review of the Application of Multiobjective Evolutionary Fuzzy Systems: Current Status and Further Directions. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 21, n. 1, p. 45–65, fev. 2013. ISSN 1063-6706.

FERNÁNDEZ, A. et al. Revisiting Evolutionary Fuzzy Systems: Taxonomy, applications, new trends and challenges. *Knowledge-Based Systems*, v. 80, p. 109–121, fev. 2015. ISSN 09507051.

FONSECA, C.; FLEMING, P. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In: FORREST, S. (Ed.). *Fifth International Conference on Genetic Algorithms (ICGA'93)*. [S.l.]: Morgan Kaufmann, 1993. p. 416–423.

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. ISSN 1432-7643.

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*, IEEE, v. 18, n. 3, p. 515–531, jun. 2010. ISSN 1063-6706.

GUILLAUME, S. Designing fuzzy inference systems from data: An interpretability-oriented review. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 9, n. 3, p. 426–443, jun. 2001. ISSN 10636706.

GUILLAUME, S.; CHARNOMORDIC, B. Fuzzy inference systems: An integrated modeling environment for collaboration between expert knowledge and data using FisPro. *Expert Systems with Applications*, Elsevier Ltd, v. 39, n. 10, p. 8744–8755, ago. 2012. ISSN 09574174.

GUILLAUME, S.; MAGDALENA, L. Expert guided integration of induced knowledge into a fuzzy knowledge base. *Soft Computing*, v. 10, n. 9, p. 773–784, jan. 2006. ISSN 1432-7643.

GUYON, I.; ELISSEEFF, A. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, JMLR.org, v. 3, p. 1157–1182, mar. 2003. ISSN 1532-4435.

HALL, M. et al. The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.*, ACM, New York, NY, USA, v. 11, n. 1, p. 10–18, 2009. ISSN 1931-0145.

HALL, M. A. *Correlation-based Feature Selection for Machine Learning*. Tese (Doutorado) — University of Waikato, Hamilton, NewZealand, 1999.

HERRERA, F. Genetic fuzzy systems: taxonomy, current research trends and prospects. *Evolutionary Intelligence*, v. 1, n. 1, p. 27–46, jan. 2008. ISSN 1864-5909.

HOLTE, R. C. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, v. 11, p. 63–91, 1993.

HOMAIFAR, A.; MCCORMICK, E. Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms. *IEEE Transactions on Fuzzy Systems*, v. 3, n. 2, p. 129–139, maio 1995. ISSN 10636706.

ISHIBUCHI, H. Multiobjective Genetic Fuzzy Systems: Review and Future Research Directions. In: *2007 IEEE International Fuzzy Systems Conference*. [S.l.]: IEEE, 2007. v. 2, n. 5, p. 1–6. ISBN 1-4244-1209-9. ISSN 1098-7584.

ISHIBUCHI, H.; MASUDA, H.; NOJIMA, Y. Selecting a small number of non-dominated solutions to be presented to the decision maker. In: *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. [S.l.]: IEEE, 2014. p. 3816–3821. ISBN 978-1-4799-3840-7.

ISHIBUCHI, H.; NAKASHIMA, Y.; NOJIMA, Y. Performance evaluation of evolutionary multiobjective optimization algorithms for multiobjective fuzzy genetics-based machine learning. *Soft Computing*, v. 15, n. 12, p. 2415–2434, nov. 2010. ISSN 1432-7643.

ISHIBUCHI, H.; NOJIMA, Y. Difficulties in choosing a single final classifier from non-dominated solutions in multiobjective fuzzy genetics-based machine learning. In: *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*. [S.l.]: IEEE, 2013. p. 1203–1208. ISBN 978-1-4799-0348-1.

ISHIBUCHI, H.; NOJIMA, Y. Repeated double cross-validation for choosing a single solution in evolutionary multi-objective fuzzy classifier design. *Knowledge-Based Systems*, v. 54, p. 22–31, 2013. ISSN 09507051.

ISHIBUCHI, H. et al. Selecting fuzzy if-then rules for classification problems using genetic algorithms. *IEEE Transactions on Fuzzy Systems*, v. 3, n. 3, p. 260–270, 1995. ISSN 10636706.

JANSSEN, J. et al. Assessment of uncertainties in expert knowledge, illustrated in fuzzy rule-based models. *Ecological Modelling*, Elsevier B.V., v. 221, n. 9, p. 1245–1251, maio 2010. ISSN 03043800.

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.

KIRA, K.; RENDELL, L. A. A Practical Approach to Feature Selection. In: *Proceedings of the Ninth International Workshop on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1992. (ML92), p. 249–256. ISBN 1-5586-247-X.

KLIR, G. J.; YUAN, B. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Upper Saddle River, NJ: Prentice Hall, 1995. 574 p. ISBN 0-13-101171-5.

KNOWLES, J. D.; CORNE, D. W. Approximating the nondominated front using the Pareto Archived Evolution Strategy. *Evolutionary computation*, MIT Press 238 Main St., Suite 500, Cambridge, MA 02142-1046 USA journals-info@mit.edu, v. 8, n. 2, p. 149–72, jan. 2000. ISSN 1063-6560.

KOHAVI, R.; JOHN, G. H. Wrappers for feature subset selection. *Artificial Intelligence*, Elsevier Science Publishers Ltd., v. 97, n. 1-2, p. 273–324, dez. 1997. ISSN 00043702.

LIU, H.; SETIONO, R. Chi2: feature selection and discretization of numeric attributes. In: *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence*. Herndon, VA: IEEE Comput. Soc. Press, 1995. p. 388–391. ISBN 0-8186-7312-5. ISSN 1082-3409.

MAMDANI, E. Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis. *IEEE Transactions on Computers*, C-26, n. 12, p. 1182–1191, dez. 1977. ISSN 0018-9340.

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. ISSN 00207373.

MILLER, G. A. The magical number seven plus or minus two: some limits on our capacity for processing information. *The Psychological Review*, v. 63, p. 81–97, 1956.

MITCHELL, T. M. *Machine Learning*. [S.l.]: McGraw-Hill, 1997. 432 p. ISBN 0070428077.

NEBRO, A. et al. A Study of Convergence Speed in Multi-objective Metaheuristics. In: RUDOLPH, G. et al. (Ed.). *Parallel Problem Solving from Nature – PPSN X SE - 76*. [S.l.]: Springer Berlin Heidelberg, 2008, (Lecture Notes in Computer Science, v. 5199). p. 763–772. ISBN 978-3-540-87699-1.

NICOLETTI, M. C.; CAMARGO, H. A. *Fundamentos da Teoria de Conjuntos Fuzzy. Série Apontamentos*. São Carlos, SP: EdUFSCar, 2004. 105 p.

NOGUEIRA, E. et al. *Perfil Andrológico de Touros Nelore Criados Extensivamente no Planalto e no Pantanal Sul-Mato-Grossense*. Corumbá, 2011. 4 p. Disponível em: <http://www.cpap.embrapa.br/publicacoes/online/CT100.pdf>.

OLIVEIRA, J. de. Semantic constraints for membership function optimization. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, IEEE, v. 29, n. 1, p. 128–138, 1999. ISSN 10834427.

PANCHO, D. P. et al. FINGRAMS: Visual Representations of Fuzzy Rule-Based Inference for Expert Analysis of Comprehensibility. *IEEE Transactions on Fuzzy Systems*, v. 21, n. 6, p. 1133–1149, dez. 2013. ISSN 1063-6706.

PANCHO, D. P.; ALONSO, J. M.; MAGDALENA, L. Quest for Interpretability-Accuracy Trade-off Supported by Fingrams into the Fuzzy Modeling Tool GUAJE. *International Journal of Computational Intelligence Systems*, Taylor & Francis, v. 6, n. sup1, p. 46–60, jun. 2013. ISSN 1875-6891.

PEDRYCZ, W.; GOMIDE, F. *An Introduction to Fuzzy Sets: Analysis and Design*. Cambridge: MIT Press, 1998. 465 p.

PULKKINEN, P.; KOIVISTO, H. A Dynamically Constrained Multiobjective Genetic Fuzzy System for Regression Problems. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 18, n. 1, p. 161–177, fev. 2010. ISSN 1063-6706.

QUINLAN, J. R. C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, p. 302, mar. 1993.

SALVADOR, D. F. et al. Associação entre o perfil andrológico e a congelação de sêmen de touros da raça Nelore aos dois anos de idade, pré-selecionados pela classificação andrológica por pontos (CAP). *Arquivo Brasileiro de Medicina Veterinária e Zootecnia*, scielo, v. 60, p. 587–593, 2008. ISSN 0102-0935.

SHUKLA, P. K.; TRIPATHI, S. P. A review on the Interpretability-Accuracy Trade-Off in Evolutionary Multi-Objective Fuzzy Systems (EMOFS). *Information (Switzerland)*, v. 3, n. 3, p. 256–277, 2012. ISSN 20782489.

SHUKLA, P. K.; TRIPATHI, S. P. On the Design of Interpretable Evolutionary Fuzzy Systems (I-EFS) with Improved Accuracy. In: *2012 International Conference on Computing Sciences*. [S.l.]: IEEE, 2012. p. 11–14. ISBN 978-1-4673-2647-6.

STAVRAKOUDIS, D. G.; THEOCHARIS, J. B.; ZALIDIS, G. C. A multistage genetic fuzzy classifier for land cover classification from satellite imagery. *Soft Computing*, v. 15, n. 12, p. 2355–2374, nov. 2010. ISSN 1432-7643.

STONE, M. Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, v. 36, n. 2, p. 111–147, 1974.

TAKAGI, T.; SUGENO, M. Fuzzy identification of system and its applications to modelling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 15, n. 1, p. 116–132, 1985.

THRIFT, P. Fuzzy logic synthesis with genetic algorithms. In: *Proc. 4th Int. Conf. Genetic Algorithms*. [S.l.: s.n.], 1991.

UCI. *UCI Machine Learning Repository*. 2015. Disponível em: <http://archive.ics.uci.edu/ml/>.

WAGSTAFF, K. Machine Learning that Matters. In: *International Conference on Machine Learning, 29*. Edinburgh, Scotland: [s.n.], 2012.

WANG, L.-X. The WM method completed: a flexible fuzzy system approach to data mining. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 11, n. 6, p. 768–782, dez. 2003. ISSN 1063-6706.

WANG, L.-X.; MENDEL, J. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics*, IEEE, v. 22, n. 6, p. 1414–1427, 1992. ISSN 00189472.

WITTEN, I. H.; FRANK, E.; HALL, M. *Data Mining: Practical machine learning tools and techniques*. 3rd. ed. San Francisco, CA: Morgan Kaufmann, 2011. 664 p. ISBN 9780123748560.

YING, H. Basic Fuzzy Mathematics for Fuzzy Control and Modeling. In: *Fuzzy Control and Modeling: Analytical Foundations and Applications*. New York: Wiley-IEEE Press, 2000. cap. 1, p. 342.

ZADEH, L. A. Fuzzy sets. *Information and Control*, v. 8, n. 3, p. 338–353, jun. 1965. ISSN 00199958.

ZADEH, L. A. Fuzzy logic, neural networks, and soft computing. *Communications of the ACM*, v. 37, n. 3, p. 77–84, mar. 1994. ISSN 00010782.

ZADEH, L. A. Soft computing and fuzzy logic. *Software, IEEE*, v. 11, n. 6, p. 48 – 56, nov. 1994. ISSN 0740-7459.

ZHANG, Q.; LI, H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, v. 11, n. 6, p. 712–731, dez. 2007. ISSN 1941-0026.

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 Syst.*, Elsevier North-Holland, Inc., Amsterdam, The Netherlands, The Netherlands, v. 159, n. 23, p. 3091–3131, 2008. ISSN 0165-0114.

ZITZLER, E.; LAUMANNS, M.; THIELE, L. *"SPEA2: Improving the strength Pareto evolutionary algorithm", TIK-Report 103*. Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute ofTechnology (ETH) Zurich, 2001. 1–21 p.