2
Sistemas Inteligentes

2.1.
Introdução

A Inteligência Computacional é uma área da ciência que busca, através de técnicas inspiradas na Natureza, o desenvolvimento de sistemas inteligentes que imitam aspectos do comportamento humano, tais como: aprendizado, percepção, raciocínio, evolução e adaptação[2].

Dentre as principais técnicas de Inteligência Computacional destacam-se as Redes Neurais, os Algoritmos Genéticos e a Lógica Fuzzy.

A técnica de Redes Neurais é inspirada nos neurônios biológicos. Assim, são modelos computacionais que procuram reproduzir características humanas, tais como: aprendizado, associação, generalização e abstração. Redes Neurais são efetivas no aprendizado de padrões a partir de dados não lineares, incompletos, com ruído ou compostos de exemplos contraditórios. Essa técnica é bastante aplicada em sistemas de previsão, reconhecimento de padrões e classificação[3][4][5][6].

A técnica de Algoritmos Genéticos (GAs - Genetic Algorithms) é inspirada nos mecanismos de evolução natural e recombinação genética. Fornece um mecanismo de busca adaptativa que se baseia no princípio Darwiniano de reprodução e sobrevivência dos mais aptos. Essa técnica é aplicada principalmente em sistemas de otimização de problemas não lineares[7][8][9][10].

A técnica de Lógica Fuzzy tem por objetivo modelar o modo aproximado de raciocínio humano, visando desenvolver sistemas computacionais capazes de tomar decisões lógicas em um ambiente de incerteza e imprecisão. A Lógica Fuzzy oferece um mecanismo para manipular informações inexatas, tais como os conceitos “muito”, “pouco”, “pequeno”, “alto”, “bom”, etc, fornecendo uma resposta aproximada para uma questão baseada em um conhecimento que é inexato, incompleto ou não totalmente confiável. Encontram-se aplicações dessa técnica em problemas de previsão, controle, classificação e reconhecimento de padrões[11][12][13][14][15].

As seções a seguir apresentam um resumo sobre as técnicas inteligentes utilizadas nesta dissertação para o desenvolvimento de uma ferramenta capaz de realizar previsões de séries temporais em diferentes horizontes, baseada na teoria de sistemas de inferência fuzzy – permitindo a interpretabilidade do modelo – com ajuste automático de parâmetros por meio de algoritmos genéticos.

2.2. Sistemas de Inferência Fuzzy

2.2.1. Introdução

Seres humanos são capazes de lidar com processos bastante complexos, baseados em informações inexatas ou aproximadas. A estratégia adotada pelos operadores humanos é também de natureza inexata e geralmente possível de ser expressa em termos linguísticos. A Teoria de Conjuntos Fuzzy e os Conceitos de Lógica Fuzzy podem ser utilizados para traduzir em termos matemáticos a informação inexata expressa por um conjunto de regras linguísticas. Se um operador humano for capaz de articular sua estratégia de ação como um conjunto de regras da forma se ... então, um algoritmo passível de ser implementado em computador pode ser construído. O resultado é um sistema de inferência baseado em regras, no qual a Teoria de Conjuntos Fuzzy e Lógica Fuzzy fornecem o ferramental matemático para se lidar com as tais regras linguísticas[16].

As primeiras aplicações bem sucedidas da Lógica Fuzzy situaram-se na área de Controle [17], mas, desde então, tem-se verificado uma utilização crescente de sistemas fuzzy em outras áreas, como classificação, mineração de dados e previsão de séries temporais.

2.2.2. Conjuntos Fuzzy

Na teoria clássica de conjuntos, o conceito de pertinência é bem definido: um elemento x simplesmente pertence ou não a um dado conjunto A, em um universo X. A função característica f_A expressa esse conceito:

$$f_A = \begin{cases} 1, se\ e\ somente\ se\ x \in A \\ 0, se\ e\ somente\ se\ x \notin A \end{cases}$$

No entanto, existem conjuntos cujo limite entre pertinência e não pertinência é vago, havendo na realidade uma transição gradual entre esses dois grupos. Por exemplo, por mais que se tente estabelecer um limite entre o conjunto de pessoas altas e o conjunto de pessoas baixas, é difícil aceitar que por causa de 1 cm uma pessoa deixa de ser baixa e passa a ser alta, e vice-versa.

Assim, a Teoria de Conjuntos Fuzzy concebida por ZADEH [18][19][20][21] propôs uma caracterização mais ampla, generalizando a função característica de modo que ela pudesse assumir um número infinito de valores dentro de um intervalo especificado. Valores maiores da função característica denotam graus maiores de pertinência. O intervalo mais usado para os valores é o intervalo unitário [0, 1]. Dessa forma, um conjunto fuzzy A em um universo X é definido por uma função de pertinência, geralmente representado da seguinte forma[13]:

$$\mu_A(x): X \rightarrow [0,1]$$

Geralmente a representação de um conjunto fuzzy é feita por um conjunto de pares ordenados:

$$A = \{\mu_A(x)/\ x\}$$

Um determinado elemento pode pertencer a mais de um conjunto fuzzy, com diferentes graus de pertinência.

As principais propriedades de um conjunto fuzzy são:

- **Altura**: é o maior grau de pertinência permitido pela função de pertinência. Um conjunto fuzzy é dito normal quando sua altura é igual a um.
- **Domínio**: universo total de valores possíveis para os elementos de um conjunto fuzzy. Um conjunto fuzzy pode apresentar um domínio aberto ou um domínio fechado.
- **Suporte**: área efetiva do domínio de um conjunto fuzzy que apresenta valores de $\mu(x) > 0$. O conjunto fuzzy cujo suporte é um único ponto X, com valor de $\mu(x) = 1$ é chamado de conjunto singleton.

Essas propriedades estão ilustradas na Figura 1.

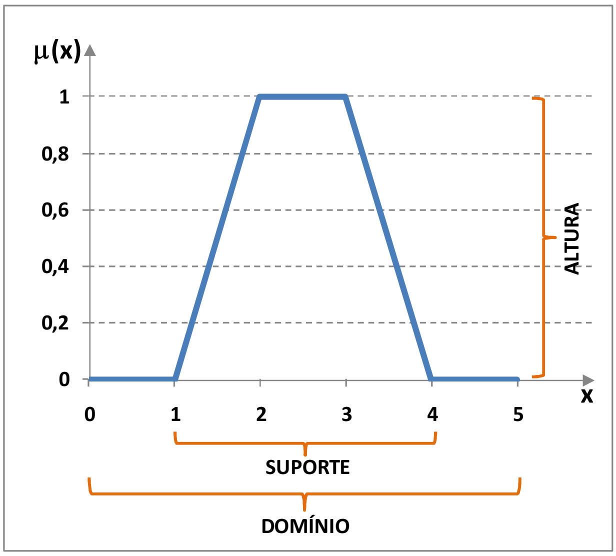

Figura 1 – Propriedades de um conjunto fuzzy

Geralmente os conjuntos fuzzy são nomeados por rótulos lingüísticos, representando conceitos inexatos. Por exemplo, a variável lingüística temperatura pode ser composta por 4 conjuntos fuzzy, com diferentes rótulos, conforme representado na Figura 2[22]:

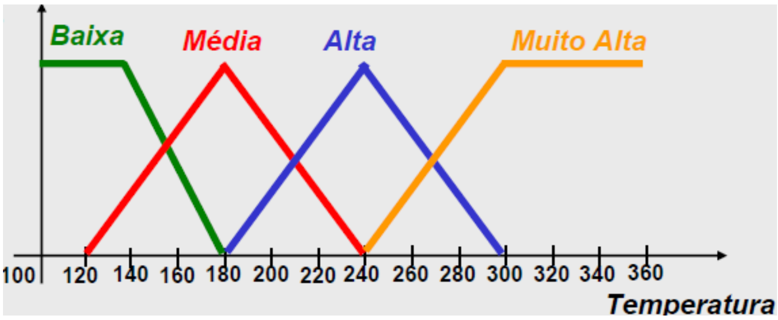

Figura 2 – Representação por variável fuzzy da temperatura

A função de pertinência que define um conjunto fuzzy pode ter diferentes formatos. Os formatos mais utilizados são o triangular, o trapezoidal e a gaussiana. As funções de pertinência contínuas podem ser definidas por funções analíticas (ex: gaussiana), enquanto as funções de pertinência descontínuas são compostas por segmentos contínuos, dando a forma desejada:

- Função de pertinência com formato triangular:

$$\mu_{triangular}(x) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{b-a}, & a < x \le b \\ \dfrac{c-x}{c-b}, & b < x \le c \\ 0, & x > c \end{cases}$$

Onde *a, b* e *c* são os parâmetros do conjunto, conforme Figura 3.

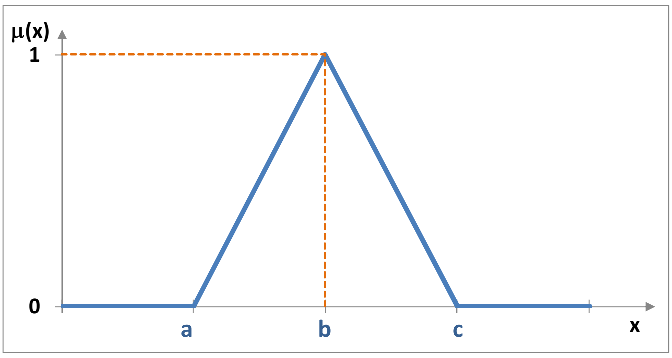

Figura 3 – Conjunto fuzzy de formato triangular

- Função de pertinência com formato trapezoidal:

$$\mu_{trapezoidal}(x) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{b-a}, & a < x \le b \\ 1, & b < x \le c \\ \dfrac{d-x}{d-c}, & c < x \le d \\ 0, & x > d \end{cases}$$

Onde *a, b, c* e *d* são os parâmetros do conjunto, conforme Figura 4.

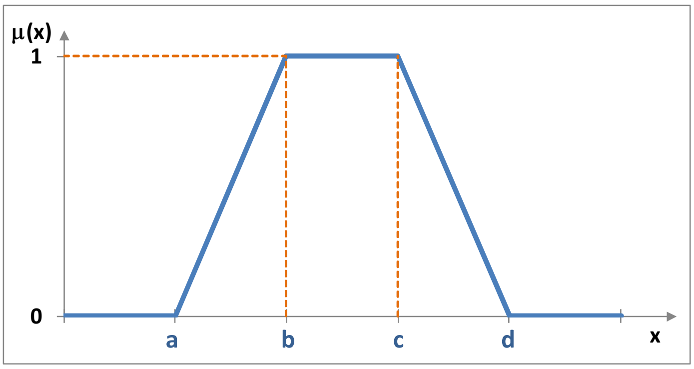

Figura 4 – Conjunto fuzzy de formato trapezoidal

As operações básicas com conjuntos fuzzy são: complemento, interseção e união.

O complemento *A'* de um conjunto fuzzy *A* é dado por:

$$\mu_{A'}(x) = 1 - \mu_A(x) \qquad \forall x \in X$$

Os operadores de interseção e de união foram inicialmente definidos de forma análoga às definições dessas operações com conjuntos precisos (crisp), utilizando o operador *min* para interseção e o operador *max* para união:

$$\mu_{A\cap B}(x) = \mu_A(x) \wedge \mu_B(x) \qquad \forall x \in X$$
$$\mu_{A\cup B}(x) = \mu_A(x) \vee \mu_B(x) \qquad \forall x \in X$$

Posteriormente, foram definidos operadores baseados nos conceitos de norma-t e conorma-t. Há inúmeros operadores norma-t e conorma-t, sendo os mais utilizados em aplicações os operadores min e produto para interseção, e max e soma-limitada para união. Maiores detalhes sobre os conceitos de norma-t e conorma-t, assim como os operadores mais utilizados, podem ser encontrados em [13][14][23][24].

2.2.3. Lógica Fuzzy

Os conceitos de lógica fuzzy foram inspirados na lógica tradicional, substituindo as funções características pelas funções de pertinência. Assim, a declaração condicional "*se x é A então y é B*" tem uma função de pertinência $\mu_{A\to B}(x,y)$ que mede o grau de verdade da relação de implicação entre x e y. Uma das formulações para $\mu_{A\to B}(x,y)$, obtida pela simples extensão de funções de pertinência bivalentes da lógica proposicional para a lógica fuzzy é[23]:

$$\mu_{A\to B}(x,y) = 1 - min[\mu_A(x), 1 - \mu_B(y)]$$

Complementando a conceituação, a regra de inferência clássica *modus ponens* é estendida para o *modus ponens generalizado:*

$$Premissa\ 1: \quad x \text{ é } A^*$$
$$Premissa\ 2: \quad Se\ x \text{ é } A\ então\ y \text{ é } B$$
$$Conclusão: \quad y \text{ é } B^*$$

No *modus ponens generalizado*, o conjunto fuzzy A* não é necessariamente o mesmo que A (antecedente da regra), assim como B* não é necessariamente o mesmo que o consequente B. Na lógica fuzzy, uma regra é disparada quando há um grau de similaridade diferente de zero entre a premissa 1 e o antecedente da regra, e o resultado é um conseqüente com grau de similaridade diferente de zero em relação ao conseqüente da regra.

Assim, pode-se afirmar que o *modus ponens generalizado* é uma composição fuzzy, onde a primeira relação fuzzy é apenas um conjunto fuzzy e a segunda relação é a relação de implicação. Dessa forma tem-se:

$$\mu_{B^*}(y) = \underbrace{sup}_{x \in A^*} [\mu_{A^*}(x) * \mu_{A \to B}(x, y)]$$

Considerando a implicação $\mu_{A \to B}(x, y) = 1 - min[\mu_A(x), 1 - \mu_B(y)]$ e conjuntos fuzzy *A* e *B* definidos por funções de pertinência triangulares, tem-se a seguinte função de pertinência para o conseqüente:

$$\mu_{B^*}(y) = 1 - min[\mu_A(x') - \mu_B(y)]$$

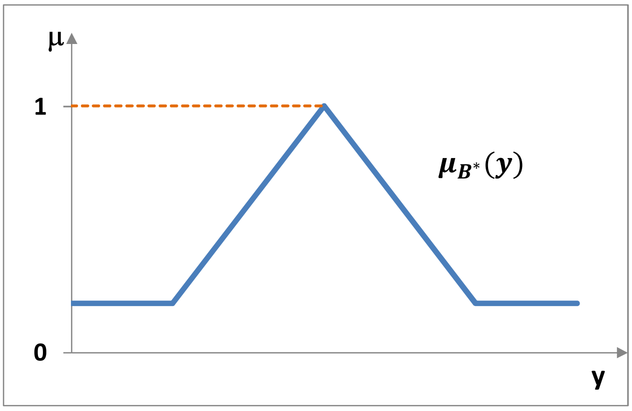

Figura 5 – Representação gráfica da função de pertinência do conseqüente com suporte infinito

Nota-se que para a implicação considerada, o resultado de uma regra específica, cujo consequente é associado a um conjunto fuzzy com suporte finito, é um conjunto fuzzy com suporte infinito, conforme ilustrado na Figura 5. Este comportamento, que se repete para outras implicações, viola o senso comum, inviabilizando aplicações práticas de engenharia.

Em virtude disto, foram definidas outras implicações, mesmo rompendo o vínculo com a lógica proposicional clássica. Assim, foram definidas as implicações *min* e *produto*, denominadas implicações de MAMDANI[25][17], ambas gerando como resultado funções de pertinência para o conseqüente com suportes finitos. O uso da implicação *min* fornece como resultado:

$$\mu_{B^*}(y) = \mu_A(x') \wedge \mu_B(y)$$

Considerando funções de pertinência triangulares para *A* e *B*, $\mu_{B^*}(y)$ terá uma forma trapezoidal de suporte finito, conforme Figura 6.

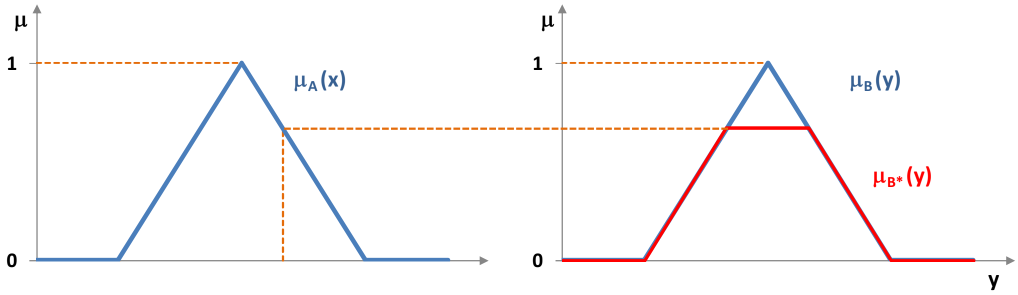

Figura 6 – Representação gráfica da função de pertinência do conseqüente com suporte finito

O grau de pertinência de *x'* em *A* estabelece o grau de ativação de uma determinada regra. Quanto mais a entrada for compatível com o antecedente da regra, mais peso terá o seu conseqüente no resultado final.

2.2.4. Sistema de Inferência Fuzzy

A estrutura de um Sistema de Inferência Fuzzy do tipo Mamdani com seus principais componentes é mostrada na Figura 7.

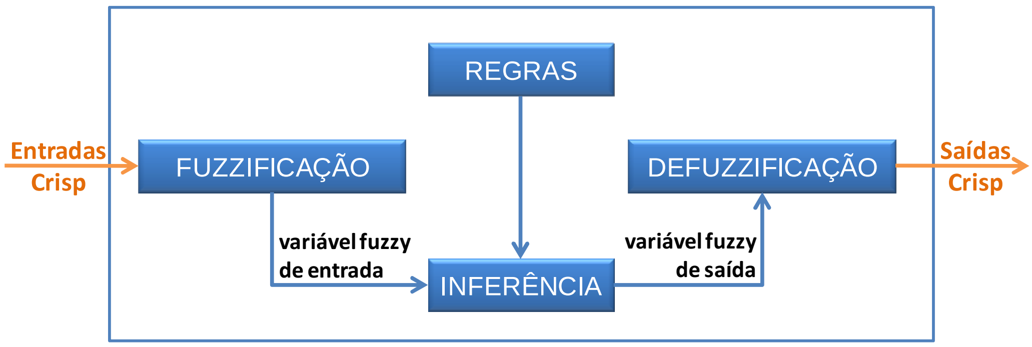

Figura 7 – Representação de um sistema de inferência fuzzy

No sistema de inferência fuzzy, o estágio de fuzzificação é responsável por traduzir as entradas crisp, resultantes de medições ou observações, em variáveis fuzzy (de entrada), compostas por conjuntos fuzzy. A definição do número de conjuntos fuzzy que compõe as variáveis de entrada – e de saída –, assim como os seus formatos, tem impacto direto no desempenho do sistema de inferência fuzzy.

As regras são fundamentais para determinar o desempenho de um sistema de inferência fuzzy. Elas são geralmente fornecidas por especialistas, por meio de sentenças lingüísticas do tipo *se... então...* No entanto, a elaboração de um banco de regras extraídas de especialistas pode não ser uma tarefa fácil, pois por mais conhecedores que eles sejam, muitas vezes é difícil traduzir todo o conhecimento em regras, mesmo utilizando variáveis lingüísticas. Alternativamente, existem métodos matemáticos para extração de regras fuzzy a partir de dados numéricos [1][11]. Estes métodos são particularmente úteis em problemas de classificação e de previsão de séries temporais.

No estágio de inferência ocorrem as operações com conjuntos fuzzy propriamente ditas: combinação dos antecedentes das regras, *implicação* e a inferência pelo *modus ponens generalizado*.

Uma vez obtido o conjunto fuzzy de saída por meio do processo de inferência, o estágio de defuzzificação é responsável por efetuar uma interpretação dessa informação, extraindo do conjunto fuzzy de saída um valor crisp de saída. Este processo é necessário porque em aplicações práticas geralmente são requeridas saídas precisas. No caso de um modelo de previsão de séries, por exemplo, em que a previsão é calculada por um sistema de inferência fuzzy, este deve fornecer ao usuário do sistema o valor previsto para o período desejado.

Existem vários métodos de defuzzificação na literatura[13][23], porém, dois dos mais utilizados são o centro de gravidade e a média dos máximos.

Com o centro de gravidade, a saída é o valor no universo que divide a área sob a curva da função de pertinência em duas partes iguais.

Já no método média dos máximos a saída é obtida tomando-se a média ponderada dos valores típicos de cada conjunto fuzzy pelo seu grau de pertinência:

$$\frac{\sum_{c=1}^{n} \mu_c \cdot z_C}{\sum_{c=1}^{n} \mu_C}$$

Onde:

μ_c: *grau de pertinencia no conjunto c*

z_c: *valor típico do conjunto c*

n: *número de conjuntos que compõe a variável de saída*

A Figura 8 exibe os parâmetros utilizados para o cálculo com dois conjuntos:

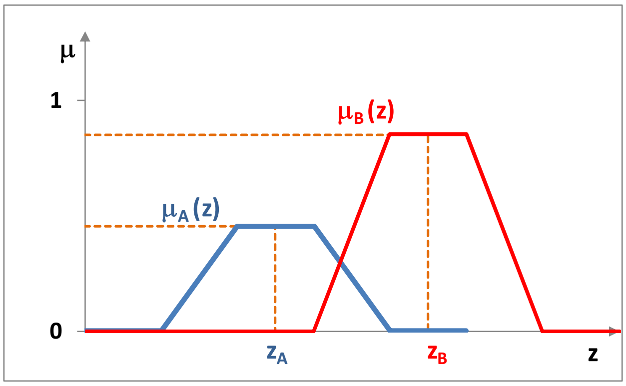

Figura 8 – Defuzzificação por média dos máximos

Existem outros tipos de sistema de inferência fuzzy além do tipo Mamdani, detalhado até aqui, dentre eles destacando-se o sistema de inferência fuzzy Takagi-Sugeno[26]. A principal diferença nesse sistema está no conseqüente, definido por uma função f – geralmente um polinômio – das variáveis dos antecedentes.

2.2.5. Extração de Regras

Em problemas reais de controle e processamento de sinais, as informações disponíveis podem ser classificadas em dois tipos: informações numéricas, obtidas por meio de medições feitas por sensores – geralmente representadas por pares entrada/saída – e informações lingüísticas, obtidas dos especialistas[11].

No entanto, cada um desses dois tipos de informação é, sozinho, incompleto. Por um lado, mesmo que um sistema seja controlado eficientemente por um especialista, certamente algumas informações serão perdidas quando os controladores humanos expressarem seus conhecimentos por meio de regras lingüísticas. Por outro lado, as informações obtidas pelos pares *input/output* passados podem não ser suficientes para garantir um bom desempenho, pois podem não abranger situações futuras.

A partir dessa base conceitual, Mendel [1][11] propôs um procedimento automático para geração de regras fuzzy a partir de dados históricos para, em seguida e se necessário, criar uma base de regras fuzzy combinada, consolidando essas regras fuzzy numéricas e as regras fuzzy lingüísticas, fornecidas por especialistas. O procedimento proposto por Mendel é composto por 5 passos, detalhados a seguir.

<u>Passo 1 – Dividir o domínio de entrada e de saída em regiões fuzzy:</u>

Seja um conjunto de pares entrada / saída composto por três variáveis, x_1, x_2 e y:

$$\left\{\left(x_1^{(1)}, x_2^{(1)}, y^{(1)}\right), \left(x_1^{(2)}, x_2^{(2)}, y^{(2)}\right), \ldots\right\}$$

Onde x_1 e x_2 são as entradas e y é a saída

Seja:

$[x_1^-, x_1^+]$: domínio de x_1

$[x_2^-, x_2^+]$: domínio de x_2

$[y^-, y^+]$: domínio de y

Cada intervalo deve ser dividido em 2N+1 regiões, sendo atribuído um conjunto fuzzy para cada uma. A Figura 9 ilustra um exemplo de divisão dos domínios de x_1, x_2 e y em cinco, sete e cinco regiões, respectivamente:

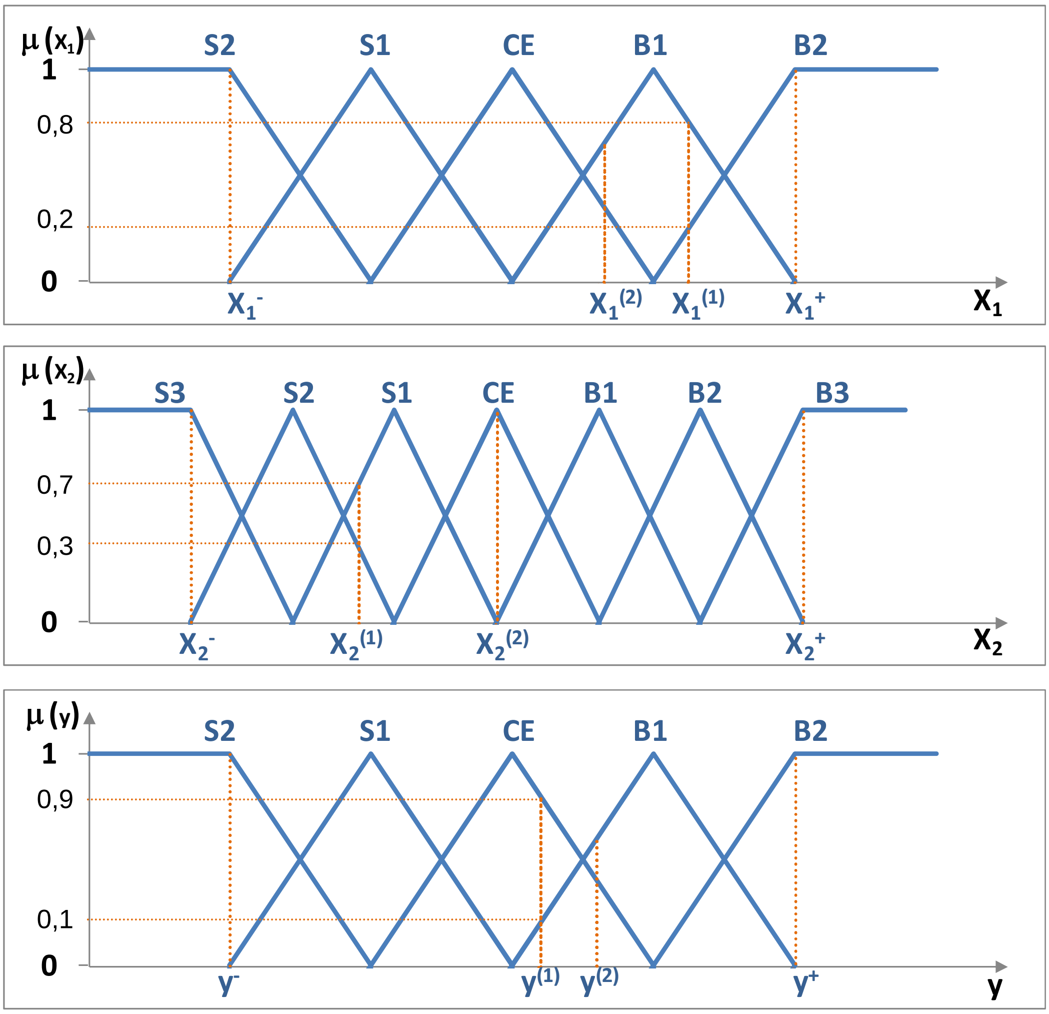

Figura 9 – Divisão dos intervalos de domínio em regiões representadas por conjuntos fuzzy

Passo 2 – Gerar regras fuzzy a partir dos dados fornecidos:

Primeiramente determina-se o grau de pertinência de cada um dos dados fornecidos $(x_1^{(i)}, x_2^{(i)}, y^{(i)})$ a cada região. Por exemplo, na figura acima, $x_1^{(1)}$ tem grau de pertinência de 0,8 em B1, de 0,2 em B2 e 0 nas demais regiões.

Em seguida, atribui-se a um valor dado a região de maior grau de pertinência. No exemplo, $x_1^{(1)}$ é considerado sendo B1, $x_2^{(1)}$ sendo S1 e $y^{(1)}$ sendo CE.

Por último, define-se uma regra a partir de cada par de dado entrada/saída. Para o par $(x_1^{(1)}, x_2^{(1)}, y^{(1)})$ tem-se a seguinte regra gerada:

$$Se\ x_1 \text{ é B1 } e\ x_2 \text{ é S1}, \qquad então\ y \text{ é CE}$$

As regras geradas dessa forma utilizam sempre o conectivo “e” no antecedente, ou seja, regras nas quais todos os antecedentes devem ter grau de pertinência diferente de zero para que o conseqüente ocorra.

Passo 3 – Atribuir um grau a cada regra gerada:

Como geralmente em aplicações práticas se trabalha com muitos pares de dados, é altamente provável a geração de regras conflitantes. Regras conflitantes são regras que apresentam os mesmos antecedentes, mas com conseqüentes diferentes. O tratamento dado para resolver essa questão consiste em atribuir um grau para cada regra e, dentro de um conjunto de regras conflitantes, aceita-se apenas aquela de maior grau, eliminando as demais. O grau de uma regra é calculado multiplicando-se o grau de pertinência de cada elemento da regra (antecedentes e conseqüente) da seguinte forma:

$$D(regra_i) = \mu_A(x_1^i) \cdot \mu_B(x_2^i) \cdot \ldots \cdot \mu_K(y^i)$$

No exemplo, a regra $Se\ x_1$ é B1 $e\ x_2$ é S1, $então\ y$ é CE apresenta o seguinte grau:

$$D(regra_1) = \mu_{B1}(x_1^1) \cdot \mu_{S1}(x_2^1) \cdot \ldots \cdot \mu_{CE}(y^1) = 0{,}8 \cdot 0{,}7 \cdot 0{,}9 = 0{,}504$$

A partir de uma análise dos pares de dados por um especialista, pode-se concluir que determinados pares são mais confiáveis e/ou úteis que outros. Nesses casos, pode-se atribuir um grau m^i a cada par de dados i, utilizando no cálculo do grau de cada regra gerada:

$$D(regra_i) = \mu_A(x_1^i) \cdot \mu_B(x_2^i) \cdot \ldots \cdot \mu_K(y^i) \cdot m^i$$

Passo 4 – Criar uma base de regras combinada:

Após a geração de todas as regras possíveis a partir dos dados numéricos – e exclusão das regras conflitantes – pode-se criar a base de regras combinada, consolidando essas regras com as regras lingüísticas criadas pelos especialistas. Para isso, assume-se que cada regra criada pelos especialistas também tem um grau associado, definido pelo próprio especialista, que traduz a importância daquela regra. Durante a consolidação dos dois tipos de regras, pode-se encontrar regras conflitantes. Nesse caso, deve-se proceder da mesma forma explicada anteriormente, ou seja, mantendo a regra de maior grau e eliminando as demais.

A base de regras do exemplo citado pode ser representada pela matriz da Figura 10, onde cada quadrado deve ser preenchido com o conseqüente:

	B3					
	B2					
	B1					
x_2	CE					
	S1				CE	
	S2					
	S3					
		S2	S1	CE	B1	B2
				x_1		

Figura 10 – Exemplo de uma base de regras fuzzy

<u>Passo 5 – Definir estratégia de defuzzificação:</u>

Para calcular o valor de saída para um dado conjunto de valores de entrada, primeiramente deve-se percorrer todas as regras geradas, calculando o grau de ativação de cada uma, por meio do uso da implicação MAMDANI, definida da seguinte forma:

$$\mu_{O^i}^{i} = \mu_{I_1^i}(x_1) \cdot \mu_{I_2^i}(x_2) \cdot \ldots \cdot \mu_{I_n^i}(x_n)$$

Onde O_i denota a região (conjunto fuzzy) do conseqüente da regra i, e I_j^i denota a região do antecedente j na regra i. No exemplo, o grau de ativação da regra 1, dado uma entrada (x_1, x_2), é calculado da seguinte forma:

$$\mu_{CE}^{1} = \mu_{B1}(x_1) \cdot \mu_{S1}(x_2)$$

Finalmente, utiliza-se, por exemplo, o método de defuzzificação por centróide para calcular a saída crisp y:

$$y = \frac{\sum_{i=1}^{K} \ \mu_{O^i}^{i} \cdot \bar{y}^i}{\sum_{i=1}^{K} \ \mu_{O^i}^{i}}$$

Onde $\bar{y}^i$ denota o centro da região O_i e K é o número de regras fuzzy presente na base de regras. O centro de uma região fuzzy é definido como o menor valor dentre aqueles que apresentam grau de pertinência igual a 1 [1].

2.2.6. Modelos Híbridos

Como dito anteriormente, a definição do número de conjuntos fuzzy que compõem as variáveis de entrada e de saída, assim como os seus formatos, tem impacto direto no desempenho do sistema de inferência fuzzy. Podem-se realizar ajustes manuais nas funções de pertinência até que se alcance um desempenho satisfatório, no entanto, é mais comum empregarem-se métodos automáticos. A integração entre sistemas de inferência fuzzy e redes neurais ou algoritmos genéticos tem se mostrado adequada para a definição de funções de pertinência [27][28] [29][30] [31][32] [33][34].

2.3. Algoritmos Genéticos

2.3.1. Introdução

Algoritmos Genéticos (GAs-Genetic Algorithms) constituem uma técnica de busca e otimização, altamente paralela, inspirada no princípio Darwiniano de seleção natural e reprodução genética[8].

Existem inúmeros problemas para os quais se deseja desenvolver um algoritmo de otimização eficiente. Para vários desses problemas é freqüentemente possível encontrar um algoritmo que ofereça uma solução ótima ou aproximadamente ótima. Porém, a maioria dos algoritmos mais eficientes requer o conhecimento do modelo matemático que representa o problema, informação muitas vezes não disponível ou difícil de obter. Um dos principais diferenciais dos Algoritmos Genéticos está exatamente na dispensabilidade dessa informação, facilitando, e algumas vezes até viabilizando a otimização de problemas complexos, com grandes espaços de busca e de difícil modelagem. Assim, GAs têm sido aplicados a diversos problemas de otimização[9][35][36][37][38].

De acordo com a teoria de C. Darwin, o princípio de seleção privilegia os indivíduos mais aptos com maior longevidade e, portanto, com maior probabilidade de reprodução. Indivíduos com mais descendentes têm mais chance de perpetuarem seus códigos genéticos nas próximas gerações. Tais códigos genéticos constituem a identidade de cada indivíduo e estão representados nos cromossomas.

Estes princípios são imitados na construção de algoritmos computacionais que buscam uma melhor solução para um determinado problema, por meio da evolução de populações de soluções codificadas em cromossomas artificiais. Em GAs um cromossoma é uma estrutura de dados que representa uma das possíveis soluções do espaço de busca do problema. Cromossomas são então submetidos a um processo evolucionário que envolve avaliação, seleção, recombinação sexual (crossover) e mutação. Após vários ciclos de evolução a população deverá conter indivíduos mais aptos, ou seja, soluções melhores segundo a função de avaliação pré-determinada[2].

O procedimento básico de um algoritmo genético é resumido na Figura 11[39]:

```
Inicio
t←1
inicializar população P(t)
avaliar população P(t)
enquanto (não condição_de_fim) faça
t←t+1
selecionar população P(t) a partir de P(t-1)
aplicar operadores genéticos
avaliar população P(t)
fim enquanto
fim
```

Figura 11 – Procedimento básico de um algoritmo genético

As seções seguintes apresentam em mais detalhes cada um dos componentes de um algoritmo genético.

2.3.2. Representação

A representação é fundamental na modelagem de um GA. Neste estágio define-se a estrutura do cromossomo, com os respectivos genes que o compõem, de maneira que este seja capaz de descrever todo o espaço de busca relevante do problema[40].

A representação do cromossoma depende essencialmente do tipo de problema. Os principais tipos de representação são os listados na Tabela 1.

Representação	Problemas
Binária	Numéricos, Inteiros
Números reais	Numéricos
Permutação de símbolos	Baseados em ordem
Símbolos repetidos	Grupamento

Tabela 1 – Principais tipos de representação

2.3.3. Decodificação

A decodificação do cromossoma consiste na reconstrução da solução real do problema a partir da solução representada pelo cromossoma. O processo de decodificação constrói a solução para que esta seja avaliada pelo problema.

2.3.4. Avaliação

A avaliação é o elo entre o Algoritmo Genético e o problema real. A avaliação é feita por meio de uma função que melhor representa o problema e tem por objetivo fornecer uma medida numérica de aptidão de cada indivíduo na população corrente, que irá dirigir o processo de busca. Funções de avaliação são específicas de cada problema.

2.3.5. Seleção e Reprodução

O processo de seleção é responsável por selecionar os indivíduos para a reprodução. A seleção é baseada na aptidão dos indivíduos: indivíduos mais aptos têm maior probabilidade de serem escolhidos para reprodução. A seleção em GAs é tipicamente implementada por uma roleta onde cada indivíduo é representado por uma fatia proporcional a sua aptidão relativa.

A forma pela qual os indivíduos são substituídos entre gerações é determinada pela técnica de reprodução empregada. Existem basicamente quatro técnicas:

- **Troca de toda a população**: todos os indivíduos da população são trocados a cada geração.
- **Troca de toda a população com elitismo**: todos os indivíduos da população são trocados a cada geração, porém o indivíduo mais apto é sempre copiado para a geração seguinte
- **Troca parcial da população (steady state):** substitui-se apenas uma parcela da população a cada geração. O número de indivíduos a serem substituídos é definido pelo GAP. Os indivíduos são ordenados pela aptidão. Os menos aptos são substituídos e os mais aptos são mantidos. Assim, dado um GAP de 30% e uma população de 100 indivíduos, a cada geração os 30 indivíduos de

menor aptidão são substituídos.

- **Troca parcial da população (steady state) sem duplicados**: técnica semelhante à anterior. No entanto, os indivíduos duplicados são descartados e substituídos. Apesar de aumentar o poder de busca paralela, implica em um maior esforço computacional para identificação dos duplicados e criação de novos indivíduos.

2.3.6.
Operadores Genéticos

Os indivíduos selecionados para reprodução são recombinados por meio de operadores de crossover e de mutação. A realização ou não do crossover e da mutação depende de parâmetros denominados, respectivamente, de taxa de crossover e taxa de mutação.

Nos operadores de crossover, pares de indivíduos são selecionados e novos indivíduos são criados a partir de troca de partes dos cromossomas dos indivíduos genitores. Dessa forma, o material genético dos indivíduos mais aptos tende a ser preservado e recombinado visando à geração de indivíduos ainda mais aptos.

Os operadores de crossover mais utilizados são:

- **Crossover de 1 ponto**: os genitores selecionados são cortados em um ponto aleatoriamente escolhido, criando dois indivíduos filhos. A Figura 12 ilustra esse processo em cromossomos de representação binária:

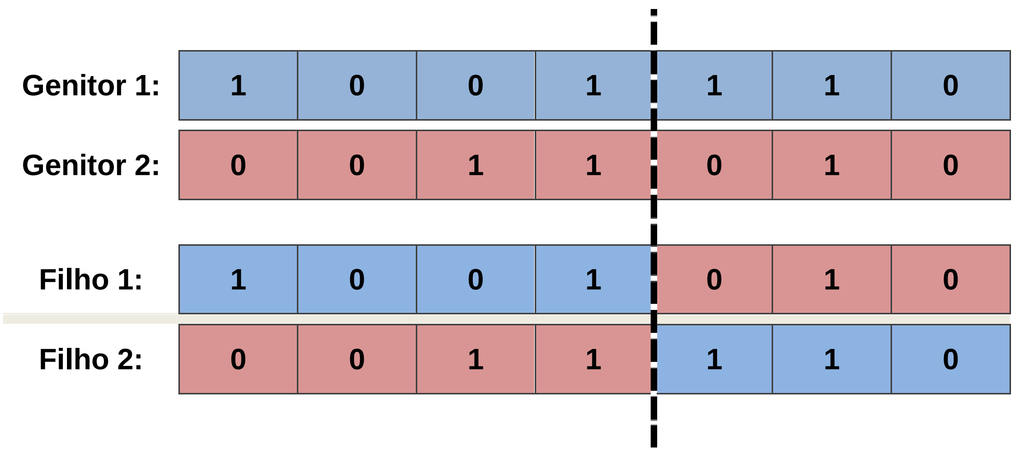

Figura 12 – Crossover de um ponto

- **Crossover de 2 pontos**: operador semelhante ao anterior, no entanto os indivíduos genitores são cortados em dois pontos aleatórios.

- **Crossover uniforme**: nesse operador, os genitores são combinados a partir de uma máscara aleatória, que define a origem de cada gene dos indivíduos gerados. A Figura 13 ilustra o processo:

Genitor 1:	1	0	0	1	1	1	0
Genitor 2:	0	0	1	1	0	1	0
Máscara:	1	1	0	0	1	0	1
Filho 1:	1	0	1	1	1	1	0
Filho 2:	0	0	0	1	0	1	0

Figura 13 – Crossover uniforme

O crossover uniforme apresenta um poder de destruição maior que o crossover de um ponto e o de dois pontos que, por sua vez, preservam os códigos compactos. Assim, o crossover uniforme deve ser aplicado em conjunto com técnicas de reprodução elitistas, como o steady state, como forma de garantir a permanência dos melhores indivíduos.

- **Crossover de média**: operador aplicado em cromossomas de representação real, onde o indivíduo filho é gerado a partir da média dos valores dos genes dos indivíduos genitores.
- **Crossover Aritmético**: operador também aplicado em cromossomas com representação real. Realiza uma combinação linear dos genitores, conforme equação abaixo:

$$F_1 = \alpha G_1 + (1-\alpha) G_2$$

$$F_2 = \alpha G_2 + (1-\alpha) G_1$$

Nos operadores de mutação, o valor de um gene de um cromossoma é trocado de forma aleatória. A mutação garante a diversidade das características dos indivíduos da população e permite que sejam introduzidas informações que não estiveram presentes em nenhum dos indivíduos. Além disto, proporciona uma busca exploratória, oferecendo oportunidade para que mais pontos do espaço de busca sejam avaliados.

Os principais operadores de mutação são:

- **Mutação por inversão de bit**: utilizado em cromossomas com representação binária, realiza a troca de 0 por 1 e vice-versa (Figura 14).

Indivíduo original:	**1**	**0**	**0**	**1**	**1**	**1**	**0**
Indivíduo após mutação:	**1**	**1**	**0**	**1**	**1**	**1**	**0**

Figura 14 – Mutação por inversão de bit

- **Mutação de real**: utilizado em cromossomas de representação real, substitui o valor de um gene por um número real aleatório.
- **Mutação creep**: utilizado em cromossomas de representação real, substitui o valor de um gene por um número real próximo (ou distante) ao valor original, possibilitando o controle do poder de dispersão do operador. Uma das formas de se implementar o operador é conforme as equações abaixo:

$$X^{t+1} = \begin{cases} X^t + \Delta(max - X^t) \\ X^t + \Delta(X^t - min) \end{cases}$$

Onde:

$$\Delta(s) = s \times rand$$

$$rand = número\ aleatório \in [0, p], \qquad p \leq 1$$

Assim, o parâmetro p determina a intensidade do ajuste. Quanto menor for o valor de p, mais próximo o novo valor será do valor original.

2.3.7.

Parâmetros da evolução

Os principais parâmetros que determinam o desempenho de um algoritmo genético são:

- **Tamanho da população**: afeta o desempenho global e a eficiência do algoritmo genético, sendo o parâmetro de definição do grau de paralelismo do algoritmo. Uma população muito pequena oferece uma pequena cobertura do espaço de busca, causando uma queda no desempenho. Uma população suficientemente grande fornece uma melhor cobertura do domínio do problema e previne a convergência prematura para soluções locais. Entretanto, uma população demasiadamente grande exige maiores recursos computacionais, levando a um maior tempo de processamento. Logo, deve-se buscar um ponto de equilíbrio no que diz respeito ao tamanho escolhido para a população.
- **Número de gerações**: total de ciclos de evolução do algoritmo genético, sendo este geralmente usado como critério de parada do algoritmo genético. Um número de gerações muito pequeno causa uma queda no desempenho: um valor grande faz necessário um tempo maior de processamento, mas fornece uma melhor cobertura do domínio do problema, evitando a convergência para soluções locais. Da mesma forma que o tamanho da população, deve-se buscar um equilíbrio na definição deste parâmetro.
- **Taxa de crossover**: define a probabilidade de haver recombinação de indivíduos após seleção. Quanto maior for esta taxa, mais rapidamente novas estruturas serão introduzidas na população. Entretanto, isto pode gerar um efeito indesejável, pois a maior parte da população será substituída, causando assim perda de variedade genética, podendo ocorrer perda de estruturas de alta aptidão e convergência a uma população com indivíduos extremamente parecidos, indivíduos estes de solução boa ou não. Com um valor baixo, o algoritmo pode-se tornar muito lento para oferecer uma resposta aceitável.
- **Taxa de mutação**: define a probabilidade de ocorrência de troca aleatória de conteúdo de um gene do cromossomo selecionado. Deve-se evitar uma taxa de mutação muito alta, uma vez que esta pode tornar a busca essencialmente aleatória, prejudicando

fortemente a convergência para uma solução ótima

A interpolação de parâmetros permite que as taxas de *crossover* e de mutação sejam variáveis ao longo dos ciclos do GA. Geralmente é desejável que a taxa de *crossover* seja maior nas primeiras gerações, nas quais os indivíduos apresentam grande variedade de material genético. À medida que as gerações vão sendo criadas, os indivíduos tendem a apresentar características cada vez mais semelhantes. Nesta fase da evolução torna-se interessante um aumento da taxa de mutação, que elevará a probabilidade de inclusão de novo material genético na população, e, conseqüentemente, de formação de melhores indivíduos[7].