

Lógica *Fuzzy*: Conceitos e aplicações

Alessandro Assi Marro, Alyson Matheus de Carvalho Souza,
Everton R. de Sousa Cavalcante, Giuliana Silva Bezerra, Rômulo de Oliveira Nunes

Departamento de Informática e Matemática Aplicada (DIMAp)
Universidade Federal do Rio Grande do Norte (UFRN)
Natal – RN – Brasil

{alessandropotiguara, alysonmatheus, evertonranielly}@gmail.com,
Giu_drawer@hotmail.com, romulo_ciencomp@yahoo.com.br

Resumo. *Este trabalho apresenta um estudo acerca da Lógica Fuzzy, sendo abordados os principais conceitos acerca dessa teoria, incluindo uma breve descrição sobre conjuntos e regras fuzzy, bem como sobre regras de inferência fuzzy. Além disso, discorre-se acerca da aplicação da Lógica Fuzzy no contexto da Inteligência Artificial e é apresentado um estudo de caso envolvendo a aplicação dessa Lógica no desenvolvimento de sistemas especialistas inteligentes. Por fim, relata-se brevemente sobre as pesquisas envolvendo Lógica Fuzzy na UFRN.*

Abstract. *This paper presents a study on Fuzzy Logic, being discussed the main concepts of this theory, including a brief description about fuzzy sets and fuzzy rules, as well as about fuzzy inference rules. In addition, it is talked about the application of Fuzzy Logic in the context of Artificial Intelligence and it is presented a case study involving the application of this Logic in the development of intelligent expert systems. Finally, it is reported briefly on the researches involving Fuzzy Logic in UFRN.*

1. Introdução

A lógica convencional ou clássica usa distinções bem definidas para separar conjuntos. Por exemplo, pode-se dizer que uma pessoa com 1,70m é considerada alta ou mediana, nunca os dois. Isso força a construção de linhas que diferenciem membros dos não membros de uma classe. Essa divisão, muitas vezes, é um processo complicado e que não reflete a realidade do problema a ser modelado. Por outro lado, a Lógica *Fuzzy* reflete a maneira como as pessoas pensam, tentando modelar o seu senso de palavras, tomada de decisão ou senso comum. Como consequência, a introdução da Lógica *Fuzzy* tem conduzido as pesquisas para sistemas inteligentes mais humanos e mais adequados à realidade.

O termo *fuzzy* em língua inglesa pode ter vários significados, que variam de acordo com o contexto de interesse, mas o conceito básico deste adjetivo passa sempre pelo vago, indistinto, incerto. As tentativas de tradução para o português ainda não são unanimidade: “nebuloso” e “difuso” são exemplos mais populares de traduções para *fuzzy*.

Nas próximas seções serão apresentados os principais conceitos concernentes à Lógica *Fuzzy* e aos conjuntos *fuzzy*, que fundamentam essa teoria, bem como as regras de inferência presentes nesse contexto. Posteriormente, são apresentadas as aplicações de maior notabilidade da Lógica *Fuzzy*, mais especificamente relacionadas ao contexto da Inteligência Artificial, designando-se uma atenção especial à utilização da Lógica *Fuzzy* como formalismo de representação do conhecimento para ser representado em sistemas especialistas, construindo, assim, os sistemas especialistas *fuzzy*. Por fim, são elencados aspectos positivos e limitações que essa lógica não-clássica apresenta.

2. Lógica *Fuzzy*

Tomando como exemplo a frase a seguir:

Embora o transformador esteja um pouco carregado, pode-se utilizá-lo por um tempo.

nessa frase existem dois termos (um pouco e um tempo) bastante subjetivos e difíceis de representar. Para um especialista que esteja conversando com outro especialista, o entendimento seria normal; no entanto, durante o processo de aquisição, fica bastante complicado para o engenheiro de conhecimento entender e representar esse tipo de conhecimento. Dessa forma, uma maneira de tentar solucionar o processo de representação de conhecimento impreciso é através da Lógica *Fuzzy*. A Lógica *Fuzzy* tem como principal objetivo a modelagem computacional do raciocínio humano, impreciso, ambíguo e vago.

A Lógica *Fuzzy* (também chamada de lógica multivalorada) foi primeiramente introduzida em 1930 pelo filósofo e lógico polonês Jan Lukasiewicz. Através do estudo de termos do tipo alto, velho e quente, ele propôs a utilização de um intervalo de valores $[0,1]$ que indicaria a possibilidade que uma declaração fosse verdadeira ou falsa. Em 1937, o filósofo Max Black propôs a idéia de que continuidade descrevia graus. Ele definiu o primeiro conjunto *fuzzy* e descreveu algumas idéias básicas de operações com conjuntos *fuzzy*. Em 1965, Lofti Zadeh publicou o artigo *Fuzzy Sets*, que ficou conhecido como a origem da Lógica *Fuzzy*. Na realidade, Zadeh redescobriu a idéia de fuzzyficação, identificou e explorou tal conceito, assim como lutou por ele. Portanto, Zadeh ficou e ainda é conhecido como o “mestre” da Lógica *Fuzzy*.

2.1. Lógica convencional \times Lógica *Fuzzy*

A teoria clássica de conjuntos permite o tratamento de classes de objetos e suas inter-relações em um universo definido. Nessa teoria, a *pertinência* de um dado elemento com relação a um conjunto refere-se ao fato de tal elemento pertencer ou não a esse conjunto. De forma ilustrativa, considere o gráfico apresentado no lado esquerdo da figura 1, que representa um exemplo típico da teoria clássica e descreve a altura de uma pessoa através de três conjuntos: *baixo*, *médio* e *alto*. Nesse exemplo, dado um elemento x qualquer, o mesmo pertencerá a um dos conjuntos do gráfico; por exemplo, se $x = 1,65$, então x pertence ao conjunto *médio* e não aos demais, ou seja, um elemento pertence ou não a um determinado conjunto e, além disso, tal elemento não pertence a mais de um conjunto.

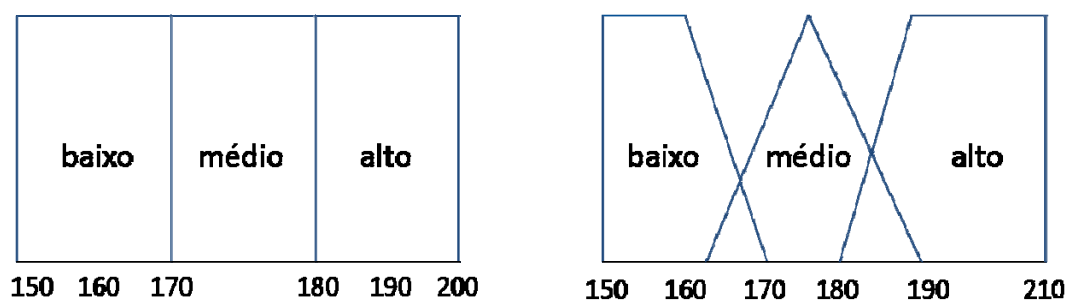


Figura 1. Representação na forma de conjuntos da altura de uma pessoa, sob o ponto de vista da Lógica convencional (à esquerda) e do da Lógica Fuzzy (à direita)

Ao contrário da Lógica convencional, a Lógica *Fuzzy* utiliza a idéia de que todas as coisas admitem (temperatura, altura, velocidade, etc.) *graus de pertinências*. Com isso, a Lógica *Fuzzy* tenta modelar o senso de palavras, tomada de decisão ou senso comum do ser humano. Ainda tomando como exemplo o gráfico da figura 1, dados dois elementos $x_1 = 1,69$ e $x_2 = 1,71$, se a Lógica clássica for utilizada, esses dois elementos pertencem a classes diferentes, x_1 pertencendo à classe *médio* e x_2 à classe *alto*. No entanto, na realidade fica difícil de dizer que uma pessoa com 1,69 m e outra com 1,71 m pertencem a classes diferentes.

Por outro lado, na Lógica *Fuzzy*, tanto o x_1 quanto o x_2 têm graus de pertinências aos conjuntos *fuzzy* definidos, que podem variar de 0 a 1. Em outras palavras, enquanto que a tomada de decisão na teoria clássica seria como a da expressão 1, a da Lógica *Fuzzy* seria como a da expressão 2, considerando um conjunto A e um elemento x com relação a esse conjunto.

$$f(x) = \begin{cases} 1 & \text{se, e somente se, } x \in A \\ 0 & \text{se, e somente se, } x \notin A \end{cases} \quad (1)$$

$$\mu(x) = \begin{cases} 1 & \text{se, e somente se, } x \in A \\ 0 & \text{se, e somente se, } x \notin A \\ 0 \leq \mu(x) \leq 1 & \text{se } x \text{ pertence parcialmente a } A \end{cases} \quad (2)$$

Dessa forma, a Lógica *Fuzzy* pode ser considerada como um conjunto de princípios matemáticos para a representação do conhecimento baseado no grau de pertinência dos termos (graus de verdade). Como pode ser observado na expressão 2, o intervalo de pertinência é $[0,1]$, onde 0 significa que um elemento não pertence a um determinado conjunto, 1 significa completa pertinência ao conjunto, e valores entre 0 e 1 representam graus parciais de pertinências. Assim, na Lógica *Fuzzy*, um elemento pertence a um conjunto com um certo grau de pertinência, fazendo com que uma determinada sentença possa ser *parcialmente verdadeira* e *parcialmente falsa*. Além do mais, um mesmo elemento pode ter graus de pertinências diferentes de 0 para mais de um conjunto *fuzzy*.

Para entender melhor o funcionamento da Lógica *Fuzzy*, ainda considerando o exemplo descrito acima, a altura de uma pessoa pode ser descrito através de conjuntos *fuzzy* (explicados com maiores detalhes na próxima seção), como mostrado no lado

direito da figura 1. Utilizando os mesmos elementos do exemplo acima, $x_1 = 1,69$ e $x_2 = 1,71$, percebe-se que ambos têm graus de pertinência 0 para o conjunto *alto* e graus de pertinência parciais para os conjuntos *baixo* e *médio*. É possível notar que, com a Lógica *Fuzzy*, não existe um limite abrupto que define os elementos que pertencem ou não a um determinado conjunto, como no caso dos conjuntos *alto*, *médio* e *baixo*. Por outro lado, os graus de pertinência dos elementos possuem variações suaves no intervalo real $[0,1]$, representando, assim, de forma mais realista, o conhecimento humano.

3. Conjuntos *fuzzy*

Como discutido anteriormente, na Lógica clássica os conjuntos são bem definidos, de modo que um elemento pertence ou não a um conjunto; se pertencer, pertence somente a um. Isso evita que ambiguidades apareçam e tornam a lógica mais simples. Ainda considerando o exemplo da utilização de conjuntos para separar pessoas pela altura, uma pessoa com 1,69m seria considerada uma pessoa de altura mediana, se assim fosse definido, estando apenas nesse conjunto e em nenhum outro; já uma pessoa com 1,71m faria parte do conjunto das pessoas altas, e somente deste. Todavia, na realidade, fica bem difícil ver que pessoas com uma diferença de altura tão mínima pertencem a conjuntos diferentes. Por outro lado, pela ótica da Lógica *Fuzzy*, ter-se-ia as duas pessoas com certo grau de pertinência aos dois conjuntos, variando entre 0 e 1, ou seja, teríamos a tomada de decisão baseada em fatores mais humanos, mais maleáveis. Assim, pode-se concluir que os conjuntos *fuzzy* que classificam os elementos de um dado universo são menos rígidos do que aqueles utilizados na teoria clássica visto que eles admitem graus parciais de pertinência.

3.1. Representação de conjuntos *fuzzy*

O primeiro passo na representação de conjuntos *fuzzy* é a escolha da função de pertinência. A escolha dessa função depende do problema a ser modelado e também da capacidade computacional disponível para processar o que se deseja. Funções não-lineares podem ser mais eficientes para problemas mais complicados, porém, elas demandam um poder computacional muito maior do que as funções lineares.

Se o universo a ser trabalhado for curto, ou contínuo, torna-se bem mais simples a aplicação de uma função para separar adequadamente os elementos em conjuntos. Um exemplo para a altura seria o apresentado na tabela 1:

Tabela 1. Altura: conjuntos *fuzzy* e graus de pertinência para alguns valores de x

Grau de pertinência	<i>alto</i>			<i>médio</i>			<i>baixo</i>		
	$x = 165$	$x = 175$	$x = 185$	$x = 145$	$x = 160$	$x = 175$	$x = 140$	$x = 150$	$x = 160$
$\mu(x)$	0	0.5	1	0	1	0	1	0.5	1

Nesse caso, tem-se o grau de pertinência de cada um sendo analisado em relação aos centímetros. Uma pessoa com menos de 165 centímetros não pode ser considerada nem um pouco alta, assim como uma pessoa acima de 160 centímetros não é nada baixa.

Como as opções para a escolha de uma função de pertinência é praticamente infinita, vale a pena sempre gastar algum tempo nessa etapa do processo e com isso escolher a função mais adequada ao problema a ser modelado. A consulta a um especialista, no caso de sistemas especialistas, é sempre uma boa idéia.

3.2. Variáveis linguísticas e modificadores

Pode-se considerar uma *variável linguística* (ou *fuzzy*) como uma entidade utilizada para representar de modo impreciso – e, portanto, linguístico – um conceito ou uma variável de um dado problema. Uma variável linguística, diferentemente de uma variável numérica, admite apenas valores definidos na linguagem *fuzzy* que está utilizando-se dela. Por exemplo:

João é alto

A variável *João* está recebendo o valor *alto*, que é um dos conjuntos *fuzzy* definidos para esta variável.

Os modificadores são termos ou operações que modificam a forma dos conjuntos *fuzzy* (ou seja, a intensidade dos valores *fuzzy*), podendo-se citar, por exemplo, os advérbios *muito*, *pouco*, *extremamente*, *quase*, *mais ou menos*, entre outros. Estes podem ser classificados em *aumentadores*, quando aumentam a área de pertinência de um conjunto *fuzzy*, ou, analogamente, *diminuidores*, quando diminuem a área de pertinência de um conjunto *fuzzy*.

3.3. Operações e propriedades em conjuntos fuzzy

Como na teoria clássica, os conjuntos *fuzzy* obedecem a certas propriedades e podem ser operados de diversas maneiras. As operações entre conjuntos são extremamente importantes para os sistemas que se utilizam dessa lógica, pois os cálculos proposicionais, por exemplo, são baseados nessas operações. O operador *AND* nada mais é que um teste se o elemento está ou não na interseção de dois conjuntos.

As operações básicas dos conjuntos *fuzzy*, definidas por Zadeh, estão definidas como segue, considerando *A* e *B* dois conjuntos *fuzzy* num universo *U*:

$$A = \{(x, \mu_A(x)) / x \in U\}, \mu_A(x) \in [0, 1]$$

$$B = \{(x, \mu_B(x)) / x \in U\}, \mu_B(x) \in [0, 1]$$

Igualdade:

$$A = B \leftrightarrow (\mu_A(x) = \mu_B(x)), \forall x \in U$$

Inclusão:

$$A \subseteq B \leftrightarrow (\mu_A(x) \leq \mu_B(x)), \forall x \in U$$

A é, então, um subconjunto *fuzzy* de *B*.

União:

$$A \text{ OR } B = A \cup B = \{(x, \max(\mu_A(x); \mu_B(x))) / x \in U\}$$

Essa definição para a união foi proposta por Zadeh na década de 1960. A maneira atualmente utilizada de definir essa operação é através de uma *norma S*, ou seja, uma família de funções que obedecem às seguintes propriedades:

- Comutatividade: $S(a, b) = S(b, a)$
- Associatividade: $S(a, S(b, c)) = S(S(a, b), c)$
- Monotonicidade: se $a \leq b$ e $c \leq d$, então $S(a, c) \leq S(b, d)$
- Coerência nos contornos: $S(a, 0) = a$ e $S(a, 1) = 1$

Assim, as funções que obedecem a S podem ser utilizadas como um operador de união entre conjuntos *fuzzy*. A união representa o operador *OR*.

Interseção:

$$A \text{ AND } B = A \cap B = \{(x, \min(\mu_A(x); \mu_B(x))) / x \in U\}$$

Essa definição também é do trabalho original, proposto por Zadeh. A definição mais utilizada atualmente é a das *normas T*, que obedecem às seguintes propriedades:

- Comutatividade: $T(a, b) = T(b, a)$
- Associatividade: $T(a, T(b, c)) = T(T(a, b), c)$
- Monotonicidade: se $a \leq b$ e $c \leq d$, então $T(a, c) \leq T(b, d)$
- Coerência nos contornos: $T(a, 0) = 0$ e $T(a, 1) = a$

As funções que obedecem a T podem ser utilizadas como operador de interseção entre os conjuntos *fuzzy*. A interseção equivale ao operador *AND*.

Complemento:

O complemento de um conjunto *fuzzy* A pode ser denotado por $\neg A$ e é expresso da seguinte forma:

$$NOT A = \neg A = \{(x, \mu_{\neg A}(x)) / x \in U \text{ e } \mu_{\neg A}(x) = 1 - \mu_A(x)\}$$

Diferença:

A diferença entre dois conjuntos é o que sobra da interseção desses conjuntos e pode ser representada da seguinte maneira:

$$A - B = (x, (\mu_{A \cap \neg B}(x))) / x \in U$$

Sejam A , B e C conjuntos *fuzzy* onde as operações de união, interseção e complemento são válidas. Estes possuem as seguintes propriedades:

Idempotência:

$$A \cup A = A \quad A \cap A = A$$

Identidade:

Sejam \emptyset o conjunto *fuzzy* vazio e \mathcal{U} o conjunto que representa o universo e considere os graus de pertinência do conjunto vazio 0 e do conjunto-universo 1:

$$A \cap \emptyset = \emptyset \quad A \cup \emptyset = A \quad A \cap \mathcal{U} = A \quad A \cup \mathcal{U} = \mathcal{U}$$

Absorção:

$$A \cup (A \cap B) = A \quad A \cap (A \cup B) = A$$

Comutatividade:

$$A \cap B = B \cap A \quad A \cup B = B \cup A$$

Associatividade:

$$(A \cap B) \cap C = A \cap (B \cap C) \quad (A \cup B) \cup C = A \cup (B \cup C)$$

Distributividade:

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C) \\ A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

Complemento duplo:

$$\neg(\neg A) = A$$

Lei transitiva:

$$\text{Se } A^{TM} B \text{ e } B^{TM} C \rightarrow A^{TM} C$$

Leis De Morgan:

$$\neg(A \cup B) = \neg A \cap \neg B \quad \neg(A \cap B) = \neg A \cup \neg B$$

4. Regras *fuzzy*

As regras *fuzzy* são regras normais utilizadas para operar, da maneira correta, conjuntos *fuzzy*, com o intuito de obter consequentes. Para criar tais regras é preciso de um raciocínio coerente com o que se deseja manusear e obter. Para isso, este raciocínio deve ser dividido em duas etapas: (1) avaliar o antecedente da regra e (2) aplicar o resultado no consequente.

Por exemplo, considerando a sentença

se x é alto, então x é pesado

seguindo os passos 1 e 2 acima, tem-se que para $x = 1,70\text{m}$, deve-se, primeiramente, verificar o grau de pertinência da entrada para o conjunto ao qual se encaixa, *alto*, que é, para este caso, $\mu(x) = 0.5$. Como o grau de pertinência da entrada x é tal, então se

deve passar este valor de pertinência para um $y = 80$ kg (por exemplo), pertencente ao conjunto *pesado*.

Para casos em que existam vários antecedentes, é preciso encontrar um grau de pertinência resultante de todos os dos antecedentes. Nos casos em que o conectivo entre os antecedentes seja *e*, deve-se utilizar métodos de combinação, contanto que o resultado não ultrapasse o valor de menor pertinência entre os antecedentes; um exemplo de método seria o mínimo das pertinências. Já nos casos em que o conectivo entre os antecedentes for *ou*, deve-se utilizar métodos de combinação, contanto que o resultado não seja menor que o maior grau de pertinência.

O raciocínio é bem mais simples para casos em que existam vários consequentes, pois o grau de pertinência resultante será o mesmo para todos os consequentes.

5. Inferência *fuzzy*

A *inferência fuzzy* é um processo de avaliação de entradas com o objetivo de, através das regras previamente definidas e das entradas, obter conclusões utilizando-se a teoria de conjuntos *fuzzy*. Esse processo pode ser feito através de *modelos de inferência*, cuja escolha deve levar em consideração o tipo de problema a ser resolvido, obtendo-se assim um melhor processamento. Existem vários métodos de inferência, mas o que geralmente é mais utilizado é o método Mamdani, que será apresentado na subseção 5.1.

5.1. Inferência *fuzzy* (difusa): Estilo Mamdani

O estilo de inferência Mamdani foi criado pelo professor Ebrahim Mamdani da Universidade de Londres (Reino Unido) em 1975 no contexto do desenvolvimento de sistemas *fuzzy* baseando-se em regras de conjuntos *fuzzy* no intuito de representar experiências da vida real. Para a construção desse sistema, foi definido um processo de raciocínio dividido em quatro passos: (1) *fuzzyficação*, (2) *avaliação das regras fuzzy*, (3) *agregação das regras fuzzy* e (4) *defuzzyficação*, cada uma delas explicadas sucintamente nas subseções a seguir.

Para ilustrar cada uma das etapas, considere a análise de riscos num projeto. Nesse domínio, identificam-se três variáveis linguísticas (as duas primeiras de entrada e a última de saída), apresentadas na tabela 2, bem como seus respectivos valores. Desta feita, quer-se estabelecer, sendo conhecidos um valor x de recurso monetário para o projeto e um número y de funcionários para trabalhar no mesmo, qual o risco z desse projeto.

Tabela 2. Variáveis e valores linguísticos na análise de riscos de um projeto

Fundos do projeto (x)	
Valor linguístico	Notação
<i>Inadequado</i>	A1
<i>Razoável</i>	A2
<i>Adequado</i>	A3

Funcionários do projeto (y)	
---------------------------------	--

Valor lingüístico	Notação
<i>Pequeno</i>	B1
<i>Grande</i>	B2

Risco do projeto (z)	
Valor lingüístico	Notação
<i>Baixo</i>	C1
<i>Normal</i>	C2
<i>Alto</i>	C3

5.1.1 Fuzzyficação

Essa etapa obtém o grau de pertinência com que cada entrada pertence a cada conjunto fuzzy. Cada uma dessas entradas foi previamente limitada no universo de discurso em questão e associada a um grau de pertinência em cada conjunto fuzzy através do conhecimento do especialista. Então para obter o grau de pertinência de uma determinada entrada *crisp* basta buscar esse valor na base de conhecimento do sistema *fuzzy*.

Para o exemplo em questão, tem-se os conjuntos *fuzzy* e graus de pertinência para cada uma das variáveis de entrada, conforme mostrado na figura 2:

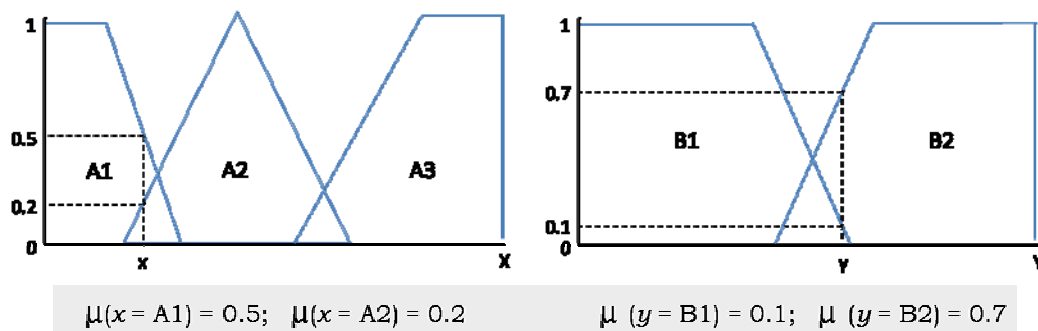


Figura 2. Fuzzyficação das variáveis lingüísticas de entrada *x* e *y*, resultando nos seus respectivos conjuntos *fuzzy* e graus de pertinência

5.1.2 Avaliação das regras *fuzzy*

Depois de obter as entradas fuzzyficadas é só aplicá-las nos antecedentes obtendo assim o valor do conseqüente para cada regra. Para um antecedente composto, os operadores *e* e *ou* são utilizados para obter um único resultado; no caso do operador *ou* é utilizada a operação de união (pega o maior grau de pertinência), e, no caso do operador *e*, é utilizada a de interseção (pega o menor grau de pertinência). Depois de obter um único valor para o antecedente é necessário obter o valor do conseqüente através de um método de correlação dos mesmos. O método mais comum é conhecido como *clipped*, onde o conseqüente é “cortado” para o nível de valor verdade do antecedente da regra avaliada, ou seja, o valor obtido é simplesmente passado para o conseqüente dessa regra.

Por exemplo, com base nos graus de pertinência e nas correlações entre as variáveis linguísticas, têm-se as regras

- 1: IF (x is A3 (0) or y is B1 (0.1)) THEN (z is C1 (0.1))
- 2: IF (x is A2 (0.2) and y is B2 (0.7)) THEN (z is C2 (0.2))
- 3: IF (x is A1 (0.5)) THEN (z is C3 (0.5))

É importante notar que, na regra 1, com a operação *ou* (*or*), tem-se que o grau de pertinência de *z* é 0.1, o maior entre os graus de *x* e de *y*; de forma similar, com a operação *e* (*and*) na regra 2, o grau de pertinência de *z* é 0.2, o menor dentre os graus de *x* e *y*. Na regra 3 foi aplicado o *clipped*: como tem-se apenas um valor na variável linguística de entrada que consta no antecedente (no caso, *x*), então o mesmo é passado para a variável linguística de saída que consta no conseqüente (no caso, *z*).

5.1.3 Agregação das regras fuzzy

Como o nome sugere, nessa etapa são agregadas todas as funções membro dos conseqüentes de cada regra em um único conjunto fuzzy. Para o exemplo em questão, considerando os conjuntos fuzzy para a variável *z* e respectivos graus de pertinência, produzidos pela aplicação das regras fuzzy 1, 2 e 3 (figura 3) a agregação desses conjuntos resulta no conjunto fuzzy da figura 4.

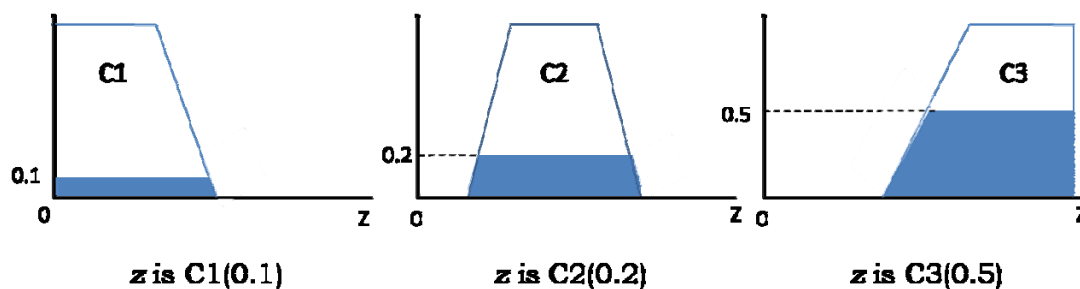


Figura 3. Conjuntos fuzzy e graus de pertinência da variável linguística de saída *z* e respectivos graus de pertinência, produzidos pela aplicação das regras fuzzy 1, 2 e 3

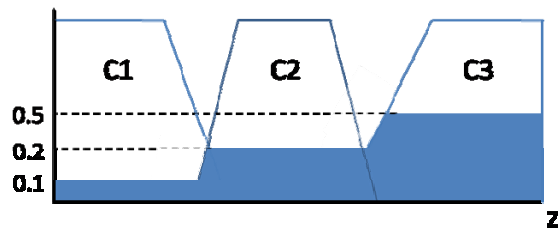


Figura 4. Conjunto *fuzzy* resultante do processo de agregação das regras *fuzzy* 1, 2 e 3

5.1.4 Defuzzyficação

Para se obter uma saída numérica é necessário defuzzyficar a saída obtida na etapa anterior. O método de defuzzyficação mais comum é a técnica do *centróide*, que obtém o ponto onde uma linha vertical divide ao meio um conjunto agregado. A fórmula matemática que obtém esse ponto é expressa da seguinte forma (expressão 3). A precisão do método depende do intervalo escolhido, quanto maior mais impreciso, porém mais rápido de calcular.

$$COG = \frac{\sum_{x=a}^b \mu(x) \cdot x}{\sum_{x=a}^b \mu(x)} \quad (3)$$

Considerando o conjunto *fuzzy* da figura 4, o resultado numérico obtido com a aplicação da técnica do centróide (*COG*) é dado por (considerando intervalos percentuais de 10%, variando de 0% a 100%):

$$COG = \frac{(0 + 10 + 20) \cdot 0.1 + (30 + 40 + 50) \cdot 0.2 + (60 + 70 + 80 + 90 + 100) \cdot 0.5}{0.1 + 0.1 + 0.1 + 0.2 + 0.2 + 0.2 + 0.5 + 0.5 + 0.5} = 67.4$$

Assim, tem-se que o risco do projeto em questão é de 67.4%.

6. Aplicações

A Inteligência Artificial é talvez a área onde a Lógica seja mais usada, visto que ela é o principal formalismo de representação do conhecimento, e, portanto, é muito útil no desenvolvimento de sistemas inteligentes, em especial os especialistas e os multiagentes, visto que, conforme comentado por Luger (2005) e Konar (2000), a representação e a inteligência são o real desafio da Inteligência Artificial. De fato, para muitos problemas reais, a imprecisão dos dados e a incerteza do conhecimento são, por natureza, parte do problema em si, e raciocinar considerando esses aspectos sem uma fundamentação adequada pode gerar inferências imprecisas.

É no campo da Inteligência Artificial onde a Lógica *Fuzzy* coloca-se como o principal instrumento para uma representação mais adequada do conhecimento (e do

próprio raciocínio), isso se devendo à sua capacidade de lidar com incertezas, raciocínio aproximado, termos vagos e ambíguos, com o que as pessoas pensam, isso tudo indo além do escopo das lógicas clássicas. Dessa forma, a Lógica *Fuzzy* permite aos sistemas computacionais inteligentes “raciocinar” considerando aspectos inerentes à incerteza e aos processos realísticos e torná-lo mais “humano”.

Abaixo segue uma lista não-exaustiva de domínios de aplicação da Lógica *Fuzzy*, no contexto da Inteligência Artificial.

- Sistemas especialistas;
- Sistemas multiagentes;
- Reconhecimento de padrões;
- Robótica;
- Sistemas de controle inteligentes;
- Sistemas de apoio à tomada de decisão;
- Algoritmos genéticos;
- *Data mining*.

McNeil e Thro (1994) relacionam algumas características de sistemas onde a aplicação da Lógica *Fuzzy* (ao que também se chama *fuzziness*, em inglês) é necessária ou benéfica:

- sistemas complexos que são difíceis ou impossíveis de modelar;
- sistemas controlados por especialistas [humanos];
- sistemas com entradas e saídas complexas e contínuas;
- sistemas que se utilizam da observação humana como entradas ou como base para regras;
- sistemas que são naturalmente “vagos”, como os que envolvem ciências sociais e comportamentais, cuja descrição é extremamente complexa.

6.1. Estudo de caso: Sistemas especialistas

Henderson (2009) define sistemas especialistas (em inglês, *expert systems*) como sistemas computacionais que empregam um determinado conhecimento (proveniente de especialistas do domínio do problema ou mesmo outras fontes de conhecimento) e regras para lidar com o mesmo no intuito de apresentar conclusões ou resolver problemas, emulando a habilidade de tomada de decisão de um especialista humano.

De acordo com Castillo e Melin (2008), sistemas especialistas que utilizam Lógica *Fuzzy* em sua concepção têm sido aplicados com sucesso a problemas de decisão, controle, diagnóstico e classificação, isso justamente porque eles são capazes de gerenciar o raciocínio complexo intrínseco a essas áreas de aplicação.

Os elementos que compõem um sistema especialista são:

- *base de conhecimento (de dados)* – banco de informações colhidas dos especialistas sobre o domínio em estudo, nela sendo representado o conhecimento que os especialistas têm sobre o domínio do problema e contendo, portanto, os elementos (dados) e formas de condução para identificação e solução de um problema;
- *mecanismo de inferência (raciocínio)* – atua como um processador, trabalhando com as informações contidas na base de conhecimento em função dos dados do problema em questão (contexto);
- internamente, existem regras utilizadas pelo mecanismo de inferência para lidar com a base do conhecimento, regras essas que são *fuzzy* (no caso de sistemas especialistas que utilizam essa Lógica).

Conforme apresentado na figura 5 a seguir, que mostra a estrutura básica de um sistema especialista que utiliza a Lógica *Fuzzy*. Como explicam Castillo e Melin (2008), em geral, diz-se que um sistema *fuzzy* implementa um mapeamento não-linear de um espaço de entrada (*input space*) para um espaço de saída (*output space*). Esse mapeamento sendo realizado pelas regras *fuzzy* na forma IF-THEN (implicação se... então...), cada uma delas descrevendo o comportamento local do mapeamento. Em particular, o antecedente da regra define uma região *fuzzy* do espaço de entrada, enquanto o conseqüente especifica a saída na região *fuzzy*.

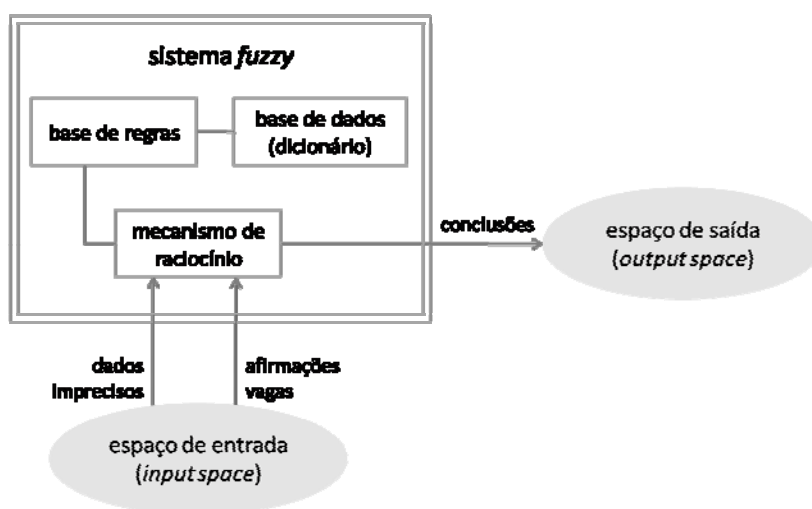


Figura 5. Estrutura básica de um sistema especialista que utiliza a Lógica *Fuzzy*

Com o objetivo de ilustrar o projeto de um sistema especialista *fuzzy*, considere o problema de operar uma central de serviços para peças extras (adaptado de Negnevitsky (2005)) com a seguinte descrição:

Uma central de serviços fornece peças de carros para seus clientes e conserta as peças defeituosas. Um cliente traz uma peça defeituosa para a central e recebe uma peça boa do mesmo tipo; a peça defeituosa é então consertada e se torna uma peça extra pronta para ser fornecida ao usuário. Se existe uma peça extra do mesmo tipo da que o cliente

trouxe, o mesmo recebe a peça extra e vai embora; em caso contrário, o cliente tem que esperar até que uma peça do mesmo tipo esteja disponível, ou seja, o cliente espera até que uma peça do mesmo tipo seja consertada. O objetivo desse sistema é dar sugestões ao gerente da central para manter o cliente satisfeito com o serviço, sempre preocupado em deixar o tempo de espera do cliente o menor possível.

Analisando a descrição acima, tem-se que o gerente dessa central de serviços para peças extras deseja definir o número de peças extras necessárias para manter o tempo de espera do cliente dentro de um intervalo aceitável.

Considerando que o desenvolvimento de um sistema especialista é incremental (evolucionário), os passos a serem seguidos – cada um deles detalhado nas subseções a seguir – são:

1. especificar o problema e definir as variáveis linguísticas;
2. definir os conjuntos *fuzzy*;
3. elicitar e construir as regras *fuzzy*;
4. codificar os conjuntos e regras *fuzzy* e os procedimentos para realizar a inferência no sistema;
5. avaliar e melhorar o sistema.

6.1.1 Especificação do problema e definição das variáveis linguísticas

O primeiro passo na construção de um sistema especialista *fuzzy* é a especificação do problema em termos de engenharia de conhecimento, ou seja, definir as variáveis de entrada e de saída, assim como seus respectivos intervalos de valores. Nesse problema, existem quatro variáveis linguísticas, a saber:

- *tempo médio de espera de um cliente (m)* – não pode exceder o limite aceitável pelo cliente e tem de ser o menor possível;
- *fator de utilização de reparo da central (p)* – média entre clientes chegando (falhas por unidade de tempo) e clientes saindo (reparos por unidade de tempo), que é proporcional ao número de empregados e ao número de peças extras disponíveis; para aumentar a produtividade da central, o gerente deve manter esse parâmetro o maior possível;
- *número de empregados (s)* – número de empregados existentes na central, ou seja, o número de pessoas disponível para o conserto das peças;
- *número de peças extras (n)* – juntamente com o número de empregados, afeta o tempo de espera, e, conseqüentemente, tem um grande impacto no desempenho da central de serviços. Aumentando s e n , obtém-se um tempo de espera pequena; entretanto, há um maior custo para empregar novos empregados e manter as peças extras nas estantes

Nesse exemplo, existem três variáveis de entrada – m , s e p – e uma de saída – n –, esta última prefigurando o ponto-chave do sistema, que é o de definir o número de peças extras necessárias para manter o tempo de espera do cliente dentro de um intervalo aceitável. A tabela 3 a seguir mostra as variáveis linguísticas e respectivos intervalos, determinados por especialistas do domínio.

Tabela 3. Variáveis linguísticas e respectivos intervalos

Tempo médio de espera (m)		
Valor lingüístico	Notação	Intervalo numérico
<i>Muito pequeno</i>	MP	[0, 0.3]
<i>Pequeno</i>	P	[0.1, 0.5]
<i>Médio</i>	M	[0.4, 0.7]

Fator de utilização (p)		
Valor lingüístico	Notação	Intervalo numérico
<i>Baixo</i>	B	[0, 0.6]
<i>Médio</i>	M	[0.4, 0.8]
<i>Alto</i>	A	[0.6, 1]

Número de empregados (s)		
Valor lingüístico	Notação	Intervalo numérico
<i>Pequeno</i>	P	[0, 0.35]
<i>Médio</i>	M	[0.3, 0.7]
<i>Grande</i>	G	[0.6, 1]

Número de peças extras (n)		
Valor lingüístico	Notação	Intervalo numérico
<i>Muito pequeno</i>	MP	[0, 0.3]
<i>Pequeno</i>	P	[0, 0.4]
<i>Pouco pequeno</i>	PP	[0.25, 0.45]
<i>Médio</i>	M	[0.3, 0.7]
<i>Pouco grande</i>	PG	[0.55, 0.75]
<i>Grande</i>	G	[0.6, 1]
<i>Muito grande</i>	MG	[0.7, 1]

6.1.2 Definição dos conjuntos *fuzzy*

Definidos os intervalos de valores para cada uma das variáveis lingüísticas, o segundo passo é definir a forma dos conceitos, materializando isso em conjuntos *fuzzy*.

Na construção e representação dos conjuntos *fuzzy*, é preciso manter interseção suficiente entre conjuntos adjacentes, de modo que a mudança entre os conceitos seja o mais suave possível. Além disso, os conjuntos *fuzzy* podem ter uma variedade de formas; entretanto, uma forma triangular ou trapezóide frequentemente fornecem uma representação adequada do conhecimento especialista e, ao mesmo tempo, simplifica significativamente o processo de computação.

As figuras 6 a 9 apresentam os diagramas dos conjuntos *fuzzy* para as quatro variáveis lingüísticas do problema:

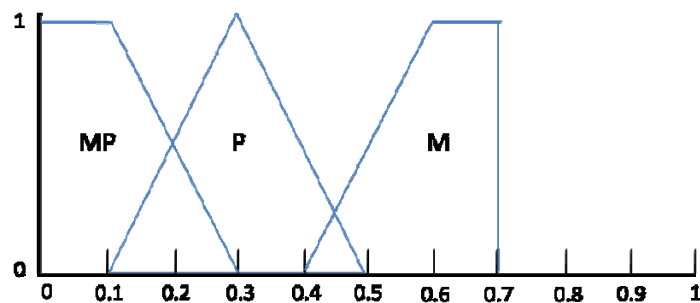


Figura 6. Diagrama dos conjuntos *fuzzy* para a variável linguística m (tempo médio de espera de um cliente)

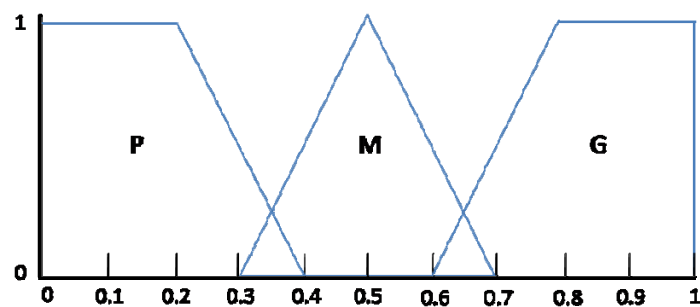


Figura 7. Diagrama dos conjuntos *fuzzy* para a variável linguística p (fator de utilização de reparo da central)

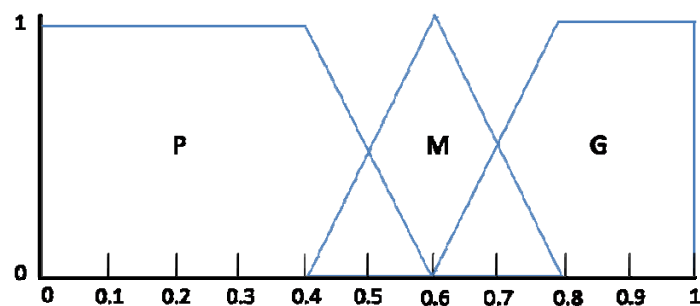


Figura 8. Diagrama dos conjuntos *fuzzy* para a variável linguística *s* (número de empregados)

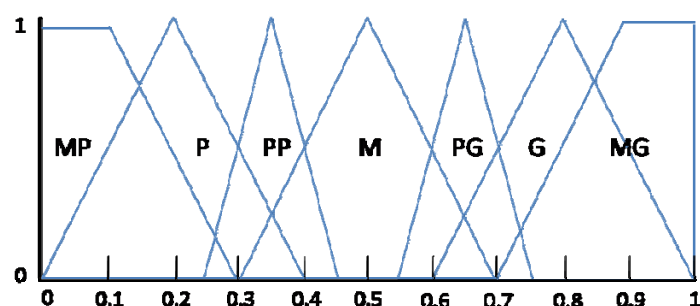


Figura 9. Diagrama dos conjuntos *fuzzy* para a variável linguística *n* (número de peças extras)

6.1.3 Elicitação e construção das regras *fuzzy*

Para construir as regras *fuzzy* relacionadas ao problema, é preciso adquirir o conhecimento envolvido, e isso é feito através de entrevistas com especialistas, no intuito de descrever como o problema pode ser resolvido usando variáveis linguísticas, ou através de outras fontes de conhecimento (e.g., livros etc.).

Considerando que há três variáveis de entrada e uma entrada de saída, é conveniente representar as regras *fuzzy* em forma de matriz (*memória associativa fuzzy* – MAF), cada coordenada representando uma variável linguística de entrada e o valor de uma célula representando um valor da variável de saída. A figura 10 mostra um exemplo de memória associativa *fuzzy* considerando as variáveis linguísticas de entrada *m* (eixo horizontal) e *s* (eixo vertical); em cada célula, têm-se os valores para a variável linguística de saída *n*.

		s		
				↑
G	M	P	MP	
M	PG	PP	S	
P	MG	G	M	
	MP	P	M	→ m

Figura 10. Memória associativa *fuzzy* (MAF) considerando as variáveis linguísticas de entrada *m* e *s*

Com base na memória associativa que relaciona as variáveis linguísticas, é possível então definir as regras *fuzzy*. Por exemplo, as regras

IF (*p* is *B*) THEN (*n* is *P*)
IF (*p* is *A*) THEN (*n* is *G*)

definem, respectivamente que, se o valor da variável *p* é *B* (fator de utilização baixo) então o valor de *n* será *P* (pequeno número de peças extras); se, por sua vez, o valor de *p* é *A* (fator de utilização alto), então o valor de *n* será *G* (grande número de peças extras). As regras *fuzzy* também podem ser mais complexas; por exemplo, as regras

IF (*m* is *MP*) AND (*s* is *P*) THEN (*n* is *MG*)
IF (*m* is *M*) AND (*s* is *G*) AND (*p* is *A*) THEN (*n* is *PP*)

definem, respectivamente, que, se o valor de *m* é *MP* (tempo médio de espera muito pequeno) e o valor de *s* é *P* (pequeno número de empregados) então o valor de *n* é *MG* (número muito grande de peças extras), como mostra a célula inferior esquerda da matriz da figura X; na outra regra, se o valor de *m* é *M* (tempo médio de espera é médio) e o valor de *s* é *G* (grande número de empregados) e o valor de *p* é *A* (alto fator de utilização), então o valor de *n* é *PP* (número muito pequeno de peças extras).

6.1.4 Codificação dos conjuntos e regras *fuzzy* e dos procedimentos de inferência

Uma vez definidos os conjuntos e as regras *fuzzy* da base de conhecimento, agora é preciso codificá-los, isto é, implementar o sistema especialista em si. Isso pode ser feito de duas maneiras: utilizando uma linguagem de programação (como C/C++, Java), que é a opção preferida dos desenvolvedores experientes e que provê uma maior flexibilidade, ou; utilizar uma ferramenta de desenvolvimento para Lógica *Fuzzy* (como, por exemplo, o *Fuzzy Knowledge Builder*TM), o que confere um rápido desenvolvimento e prototipagem de um sistema especialista *fuzzy*.

6.1.5 Avaliação e melhoria do sistema

Por fim, o último (e mais trabalhoso) passo consiste em verificar se o sistema atende aos requisitos especificados inicialmente no projeto, avaliando-se o mesmo e definindo-se possíveis melhorias que porventura são pertinentes de serem feitas. Nesta fase, a opinião do especialista é importante, pois ele pode estar ou não satisfeito com o sistema.

Embora efetuar melhorias no sistema demanda mais tempo e esforço que determinar os conjuntos *fuzzy* e construir as regras *fuzzy*, as melhorias que podem ser feitas no intuito de aumentar o desempenho de um sistema especialista *fuzzy* pode envolver um conjunto de ações:

1. Rever o modelo das variáveis de entrada e de saída e se é necessário redefinir seus intervalos;
2. Rever os conjuntos *fuzzy* e se é necessário definir conjuntos adicionais com relação ao domínio (universo do discurso); nesse ponto, é importante

ressaltar que um número maior de conjuntos *fuzzy* torna o funcionamento do sistema mais preciso;

3. Fornecer sobreposição suficiente entre conjuntos *fuzzy* adjacentes (recomendado: 25% a 50% nas bases);
4. Rever as regras *fuzzy* e se é necessário adicionar novas regras à base de regras;
5. Verificar se a inclusão de modificadores nos conjuntos *fuzzy* pode melhorar o desempenho do sistema;
6. Ajustar o peso de execução das regras (a maioria das ferramentas para Lógica *Fuzzy* permite o controle da importância das regras);
7. Rever a forma dos conjuntos *fuzzy*, se é necessário alterá-la no intuito de prover uma maior precisão.

7 Considerações finais

Como dito anteriormente, a principal vantagem de se utilizar da Lógica *Fuzzy* deve-se a sua capacidade de lidar com incertezas, raciocínio aproximado, termos vagos e ambíguos, o que não é possível de se fazer com as lógicas clássicas. O raciocínio humano envolve todos esses elementos tratados pela Lógica *Fuzzy*; por isso ela é de suma importância no contexto da Inteligência Artificial, que procura representar o raciocínio, conhecimento humano da forma mais realística possível. Além disso, conforme apontado por Munkata (2008), para problemas difíceis, métodos não-*fuzzy* convencionais normalmente são caros e dependem de aproximações matemáticas (como, por exemplo, a linearização de problemas não-lineares), o que pode comprometer o desempenho dessas soluções.

Embora apresente muitas vantagens, a Lógica *Fuzzy* também apresenta algumas limitações relevantes, a saber (Munkata, 2008):

- *Estabilidade*. Sistemas *fuzzy* são estáticos, não sendo capazes de se adaptar a contextos extremamente dinâmicos.
- *Falta de capacidade de aprendizagem*. Pelo fato de serem estáticos, sistemas *fuzzy* não conseguem “aprender”.
- *Definir boas funções de pertinência e regras fuzzy não são tarefas fáceis*. Questões como o porquê de um determinado sistema especialista *fuzzy* precisa de tantas regras, ou quando um desenvolvedor pode parar de adicionar mais regras não são facilmente respondidas.
- *A verificação e validação de um sistema especialista fuzzy geralmente requer testes extensivos*. Muitos desenvolvedores não conseguem fazer isso ou mesmo as circunstâncias não permitem esses inúmeros testes.

Para superar alguns desses problemas, principalmente no tocante à estabilidade e a falta de capacidade de aprendizagem, as atuais pesquisas envolvendo a Lógica *Fuzzy* e a sua aplicação em sistemas inteligentes consistem na combinação (hibridização) de sistemas *fuzzy* com outras técnicas a fim de prover adaptabilidade, técnicas essas como

as redes neurais [artificiais] (*neuro-fuzzy systems*, em inglês) e os algoritmos genéticos. Como coloca Coppin (2004), esses sistemas híbridos que integram esse aspecto adaptativo aprendem a lidar com situações que não tinham sido previstas anteriormente nas fases de projeto do sistema, e, em casos extremos, eles também aprendem a sobreviver quando o ambiente no qual eles operam sofre mudanças.

7.1. Pesquisas em Lógica *Fuzzy* na UFRN

Na Universidade Federal do Rio Grande do Norte (UFRN) têm-se desenvolvido pesquisas envolvendo a Lógica *Fuzzy* nos projetos apresentados na tabela 4 e sucintamente detalhados nas subseções a seguir.

Tabela 4. Projetos de pesquisa envolvendo Lógica *Fuzzy* na UFRN

Título do projeto	Coordenador do projeto
Aplicando técnicas da Inteligência Computacional em um sistema baseado em agentes para a Classificação de Biométricas	Prof. Dr ^a . Anne Magály de Paula Canuto (Professora adjunto 3 do Departamento de Informática e Matemática Aplicada – DIMAp)
Teoria dos autômatos <i>fuzzy</i> : Linguagens formais e Computabilidade	Prof. Dr. Benjamín René Callejas Bedregal (Professor associado I do Departamento de Informática e Matemática Aplicada – DIMAp)
Uso da Lógica <i>Fuzzy</i> no apoio à tomada de decisão na gestão de custos no CDI do HUOL	
Fundamentos e aplicações da Lógica <i>Fuzzy</i> intervalar	

7.1.1 Aplicando técnicas da Inteligência Computacional em um sistema baseado em agentes para a Classificação de Biométricas

Este projeto visa à utilização de técnicas da inteligência computacional em um sistema multiagente aplicado a classificação de padrões (*ClassAge*), mais especificamente para o reconhecimento de biométricas. O sistema *ClassAge*, também conhecido como *NeurAge*, é um sistema multiagente em que vários agentes buscam um consenso a respeito de um padrão de entrada fornecido ao sistema. O presente projeto visa prover algumas melhorias ao sistema *ClassAge*, através da utilização de técnicas da Inteligência Computacional. Basicamente, estas melhorias seriam basicamente (1) a proposta de métodos de negociação para o sistema, assim como (2) a utilização de algoritmos de otimização para otimizar o projeto dos métodos de classificação dos agentes e (3) a utilização de conhecimento da Lógica *Fuzzy* tanto no processo de tomada de decisão dos agentes quanto no processo de negociação entre os agentes. Por fim, pretende-se desenvolver um ambiente multiagente para classificação de padrões robusto e eficiente que possa ser amplamente utilizado na classificação de padrões, inclusive o reconhecimento de biométricas.

7.1.2 Teoria dos autômatos *fuzzy*: Linguagens formais e Computabilidade

Basicamente, a proposta de pesquisa é estudar a relação entre a Lógica *Fuzzy* com a teoria das linguagens formais e da Computabilidade. Em linguagens formais *fuzzy* pretende-se estudar classes de linguagens *fuzzy*, tanto do ponto de vista de gramáticas como de autômatos, que tenham sido pouco ou ainda não estudadas. Neste sentido, tem-

se, por exemplo, as classes das linguagens lineares *fuzzy*, linguagens sensíveis ao contexto *fuzzy* e hierarquias de linguagens livres de contexto *fuzzy* baseadas em medidas de não-determinismo.

Na pesquisa a realizar, pretende-se analisar as propriedades de fecho dessas classes com respeito a operações usuais de linguagens tais como união, interseção, complemento, etc. assim como formas normais, comparar com sua versão clássica, etc., sendo vistas também algumas aplicações de gramáticas *fuzzy* em teoria dos jogos evolutivos. No contexto da Computabilidade *fuzzy*, a pesquisa leva ao estudo das versões *fuzzy* de funções parciais recursivas e de máquinas de acesso aleatório (modelos RAM) e à comparação com alguns dos modelos de máquinas de Turing *fuzzy*, de modo a ter modelos para lidar com a Computabilidade em mundos ideais (contínuos) baseadas em representações finitas (pontos flutuantes) dos números reais com graus de pertinências (graus de incerteza) que reflitam quão longe está essa representação do valor ideal. Assim, esta abordagem trata sistemas *fuzzy* como uma alternativa à Matemática Intervalar.

7.1.3 Uso da Lógica *Fuzzy* no apoio à tomada de decisão na gestão de custos no CDI do HUOL

Atualmente no Centro de Diagnóstico por Imagem (CDI) do Hospital Universitário Onofre Lopes (HUOL), em Natal-RN, por não possuir um sistema de custos, os preços dos procedimentos são estabelecidos pela tabela do Sistema Único de Saúde (SUS) e não com base nos custos reais hospitalares levantados para cada procedimento. Por outro lado, apurar de modo exato o custo final de um procedimento é em geral impossível, uma vez que depende de uma grande quantidade de fatores, nem sempre plausíveis de serem determinados precisamente e que em geral variam caso a caso em função da gravidade do diagnóstico do paciente assim como de outros fatores. Assim, uma abordagem usando teorias que lidem com imprecisões (Lógica *Fuzzy*, Matemática Intervalar e Probabilidade), seria útil para estabelecer o custo individual e “médio” (com uma margem de erro controlada) do procedimento e no auxílio da tomada de decisões na gestão de custos.

O objetivo do projeto de pesquisa é o desenvolvimento de dois sistemas computacionais, um de custos e outro de apoio à tomada de decisão, que se complementem e que permitam auxiliar a ação gerencial, não só no monitoramento do custeio apurado, mas principalmente para definir linhas de ações gerenciais mais amplas, na medida em que possibilitem identificar caminhos estratégicos para melhorar o desempenho custo/receita dos procedimentos realizados no CDI do HUOL. O segundo destes sistemas usaria a Lógica *Fuzzy* para identificar os procedimentos mais rentáveis, aqueles cujos custos precisam ser analisados e controlados com maior cuidado e verificar sua viabilidade econômica. Esse sistema também poderá sugerir eventuais ações que conduzam à melhoria do desempenho de cada procedimento para no mínimo (sempre que possível) atingir o ponto de equilíbrio.

7.1.4 Fundamentos e aplicações da Lógica *Fuzzy* intervalar

Tanto a Lógica *Fuzzy* quanto a Matemática Intervalar são teorias que lidam com incertezas e a inexatidão do conhecimento e raciocínio humano, existindo uma grande

sinergia entre elas. Um dos relacionamentos que ocorrem entre essas teorias é a chamada Lógica Fuzzy Intervalar (LFI), que usa intervalos como graus de pertinência de conjuntos *fuzzy* com o objetivo de tratar a incerteza associada aos computadores digitais. No mundo científico, esta área tem tido ultimamente grande destaque, tanto na comunidade de Lógica *Fuzzy* quanto na de Matemática Intervalar. No Brasil, no entanto, esta área tem tido pouca difusão e desenvolvimento e, nessa perspectiva, esse projeto configura-se como uma tentativa de fortalecer as pesquisas em LFI tanto nos aspectos formais como aplicados. A idéia é concentrar esforços para mostrar a sua potencialidade na resolução de problemas e disseminar esta teoria na comunidade científica, consolidando, assim, essa linha de pesquisa dentro do Brasil.

Referências

- Castillo, Oscar; Melin, Patricia. (2008) *Type-2 Fuzzy Logic: Theory and applications*. Springer-Verlag Berlin Heidelberg.
- Coppin, Ben. (2004) *Artificial Intelligence illuminated*. Jones and Bartlett Publishers.
- Hayes-Roth, F.; Waterman, D.A.; Lenat, D.B. (1983) *Building expert systems*. Addison-Wesley.
- Henderson, Harry. (2009) *Encyclopedia of Computer Science and Technology*. Facts on File.
- Konar, Amit. (2000) *Artificial Intelligence and soft computing: Behavioral and cognitive modeling of the human brain*. CRC Press.
- Konar, Amit. (2005) *Computational Intelligence: Principles, techniques and applications*. Springer-Verlag Berlin Heidelberg.
- Luger, George F. (2005) *Artificial Intelligence: Structures and strategies for complex problem solving – 5th ed*. Pearson Education/Addison Wesley.
- Munakata, Toshinori. (2008) *Fundamentals of the new Artificial Intelligence: neural, evolutionary, fuzzy and more – 2nd ed*. Springer-Verlag London.
- McNeill, F. Martin; Thro, Ellen. (1994) *Fuzzy Logic: A practical approach*. AP Professional/Academic Press.
- Negnevitsky, Michael. (2005) *Artificial Intelligence: A guide to Intelligent Systems – 2nd ed*. Pearson Education/Addison Wesley.
- Partridge, Derek. (1998) *Artificial Intelligence and Software Engineering: Understanding the promise of the future*. Glenlake Publishing Company/American Management Association (AMACOM).
- Ross, Timothy J. (2004) *Fuzzy Logic with Engineering applications – 2nd ed*. John Wiley & Sons.
- Sivanandam, S. N.; Sumathi, S.; Deepa, S. N. (2007) *Introduction to Fuzzy Logic using MATLAB*. Springer-Verlag Berlin Heidelberg.