REVISTA INTERDISCIPLINAR DE PESQUISA EM ENGENHARIA

CILAMCE 2016
XXXVII IBERIAN LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGINEERING
BRASÍLIA - DF - BRAZIL

O USO DA LÓGICA *FUZZY* PARA SUPORTE À TOMADA DE DECISÕES EM MODELAGENS DE SISTEMAS MULTIAGENTES

Henrique Costa Braga

Gray Farias Moita

Paulo Eduardo Maciel de Almeida

bragaseg@yahoo.com.br

gray@dppg.cefetmg.br

pema@lsi.cefetmg.br

Programa de Pós-Graduação em Modelagem Matemática e Computacional

Av. Amazonas 7675, Nova Gameleira, Belo Horizonte, MG, CEP 30510-000, Brasil

Resumo. *A lógica fuzzy pode ser aplicada na emulação do processo de tomada de decisão humana em modelagens de sistemas multiagentes. Entretanto estas aplicações se limitam quase que exclusivamente ao contexto acadêmico, enquanto outras formas de utilização da lógica fuzzy já se encontram empregadas efetivamente em aplicações profissionais. Um dos fatores que pode estar contribuindo para isto é a sua velocidade de processamento. Neste trabalho vamos apresentar algumas técnicas que podem ser usadas para se acelerar o tempo de processamento fuzzy. A ideia central está em se captar todo o conhecimento fuzzy de um processamento padrão, e então o simular por outra forma de representação que apresente resultados similares mas que tenha um tempo de processamento computacional muito inferior – no caso a Matriz Fuzzy. Um exemplo prático de uma modelagem multiagente no contexto da simulação da evacuação de um ambiente construído será apresentado mostrando a equivalência dos resultados e resultando numa redução no tempo de processamento de cerca de 140 vezes sem perda da precisão dos resultados.*

Palavras-chave: *Lógica Fuzzy, Matriz Fuzzy, Sistemas complexos, Modelagem, Sistemas multiagentes, Inteligência computacional.*

1 INTRODUÇÃO

A lógica *fuzzy* tem um vasto campo de aplicação, possuindo diversas características interessantes como a possibilidade de se contextualizar um dito conceito especialista onde variáveis individuais podem não ser definidas em termos exatos (ROJAS, 1996; OLIVEIRA et al, 2007; MARRO et al., 2010). Atualmente, entre as principais aplicações tradicionais da lógica *fuzzy* se tem a elaboração de sistemas de apoio à tomada de decisão, aproximações de funções e controle de processos industriais por modelos como o de Mamdani ou Sugeno (BIONDE NETO et al., 2006; SIMÕES, SHAW, 2007).

Também existem trabalhos que já utilizaram com sucesso a lógica *fuzzy* para a emulação do processo de tomada de decisão humana em simulações de sistemas complexos multiagentes (BOULMAKOUL, MANDAR, 2011; BRAGA et al., 2014). Em sistemas complexos emergem fenômenos, sem a intervenção de um controlador central, como resultado da interação entre os agentes (BOCCARA, 2004). Entretanto estes trabalhos se limitam quase que exclusivamente ao contexto acadêmico, enquanto outras formas de emprego da lógica *fuzzy* já se encontram a tempo empregado efetivamente em aplicações profissionais. A simulação multiagente no contexto não acadêmico se faz atualmente por outras técnicas que normalmente não empregam a lógica *fuzzy* (THOMPSON, MARCHANT, 1995; PELECHANO et al., 2008).

Um dos fatores que pode estar contribuindo para a dificuldade da aplicação da lógica *fuzzy* na modelagem multiagente fora do contexto acadêmico é a sua velocidade de processamento. Um sistema *fuzzy* engloba uma série de operações em todas as suas etapas, tais como a conversão escalar para *fuzzy*, todo o processamento da máquina de inferência e sua associação ao banco de regras, assim como a conversão final de *fuzzy* para escalar (ALMEIDA, EVSUKOFF, 2003). Na Figura 1 se apresenta um diagrama típico para um sistema de processamento de conhecimento *fuzzy* que utiliza o modelo de inferência Mamdani.

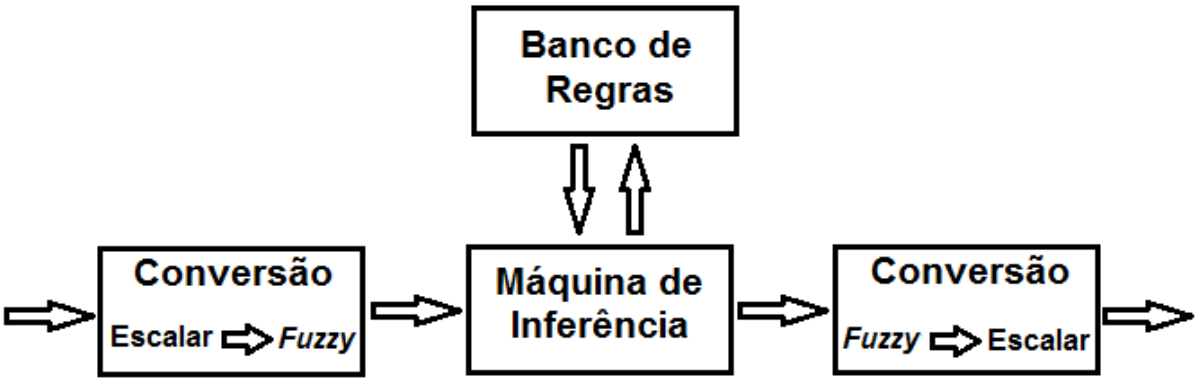


Figura 1. Diagrama típico de um sistema *fuzzy* que utiliza a metodologia Mamdani

Fonte: baseado em Almeida e Evsukoff (2003).

Todo este processo tem certa sofisticação, portanto um custo computacional. Considerando uma aplicação em um controle de processo, por exemplo, o tempo computacional gasto para um ciclo completo de uma tomada de decisão *fuzzy* pode ser na prática considerado desprezível, portanto não é limitador do seu uso. Entretanto uma

modelagem multiagente pode envolver milhares de agentes, ou mais, onde cada agente pode precisar realizar individualmente vários ciclos de processamentos *fuzzy* completos a cada segundo. Neste caso o tempo de processamento se torna um ponto negativo à utilização desta ferramenta fora do contexto acadêmico.

Neste trabalho são apresentadas técnicas que podem ser usadas para se acelerar o tempo de processamento *fuzzy*. A ideia central está em se captar todo o conhecimento *fuzzy* de um processamento padrão, e então o simular por outra forma de representação que apresente resultados similares, mas que tenha um tempo de processamento computacional muito inferior. Assim a lógica *fuzzy* poderá vir a ser utilizada na modelagem multiagente, mesmo que indiretamente, sem prejuízo da qualidade, mas com também alta velocidade de processamento. Um exemplo prático de uma modelagem multiagente no contexto da simulação da evacuação de um ambiente construído será apresentado mostrando a equivalência dos resultados e confirmando uma grande redução no tempo de processamento.

2 FUNDAMENTAÇÃO

2.1 Descrição do sistema *fuzzy* para simulação multiagente

Como exemplo de uma simulação multiagente em um sistema complexo se tem um sistema *fuzzy* completo utilizado para simulação da movimentação de pessoas em um ambiente construído. Esse sistema foi implementado computacionalmente utilizando o Matlab sendo denominado Fuga v. 1.0 (BRAGA et al., 2014).

2.1.1 Variáveis de entrada

Neste sistema, as variáveis de entrada são aqueles aspectos ergonômicos, selecionados entre as grandezas que influenciam na movimentação, que se irá efetivamente utilizar para a tomada de decisões. Possibilitados pela lógica *fuzzy*, pode-se também simplesmente criar livremente novas grandezas, desde que possam ser de quaisquer formas relacionadas com as variáveis que influenciam na movimentação de pessoas. Estas grandezas podem ter qualquer natureza (quantitativa ou qualitativa, material ou mental) e podem abordar qualquer aspecto. No Fuga as grandezas de entrada do sistema *fuzzy* são DA, EP, RP, NE, EI e EA, onde:

- DA (Distância Aparente): A DA é obtida comparando-se o valor da matriz distância aparente da casa do autômato com o valor da matriz distância aparente da casa vizinha em análise. Os valores de DA então serão:

DA=0; se na comparação a distância aparente aumentar (ou piorar),

DA=0,5; se mantiver a mesma distância aparente, e

DA=1; se a direção verificada indica um casa com menor distância aparente que a atual ocupada.

- EP (Efeito Parede): está relacionado com a busca pela zona de conforto (PAN, 2006). Associado ao modelo existe um algoritmo que verifica se ao redor da casa ocupada pelo autômato existe uma parede. Assim, os valores de EP serão:

EP = 0; se casa vizinha do autômato em verificação for também vizinha imediata a uma parede (grande efeito repulsivo);

EP=0,5; se for de dois elementos de matriz afastada a alguma parede; e

EP = 1; se posição for de três elementos de matriz ou mais afastada de alguma parede (nenhum efeito repulsivo).

- RP (Rota Preferencial): a RP é obtida comparando-se a direção de movimentação indicada pelo valor da matriz rota de fuga da casa do autômato, com a efetiva direção de movimentação que o autômato faria se movimentasse para esta casa vizinha em verificação. Os valores de RP então serão:

RP=0; se direção de movimentação do autômato para casa vizinha em análise for oposta a direção indicada pela casa da matriz rota de fuga ocupada pelo autômato,

RP=0,5; se for ortonal a indicada, e

RP=1; se a direção verificada coincidir com a direção indicada.

- NE (Nível de Estresse): antes de cada simulação (parâmetro de entrada) determinamos o nível de estresse dos autômatos (baixo, médio, alto ou muito alto). Assim, baseado na hipótese do U-invertido (STAAL, 2004) e dependendo deste nível de estresse previamente definido, os valores de NE serão:

NE = 0; para situações de médio estresse,

NE = DA; para situações de baixo ou alto estresse, e

NE= *rand* (1); para situações de muito alto estresse.

Adicionalmente o NE também, independente do seu valor inicial, possuirá uma componente aleatória. Assim, a cada iteração o NE terá uma probabilidade (geralmente de 5%), de possuir um valor aleatório.

- EI (Efeito Inércia): o EI é obtido comparando-se a direção de movimentação que o autômato faria se movimentasse para esta casa vizinha em verificação com a última direção de movimentação. Os valores de EI então serão:

EI = 0; se em direção em verificação for oposta à última realizada,

EI=0,5; se for ortogonal ao último realizado, e

EI = 1; se movimento mantiver a mesma direção do último realizado.

- Efeito Autômato (EA): associado ao modelo existe um algoritmo que verifica se ao redor da casa ocupada pelo autômato existe outro autômato. Assim, os valores de EA serão:

EA = 0; se houver um autômato vizinho até seis elementos de matriz de distância na direção considerada; e

EA = 1; se a direção estiver desobstruída por autômatos.

2.1.2 Variável de Saída

A cada iteração existem até quatro possibilidades de movimentação para cada autômato (em uma vista de topo do ambiente a movimentação pode acontecer para a esquerda, direita, acima ou abaixo). O sistema *fuzzy* irá valorar a cada iteração independentemente para cada autômato estas quatro possibilidades de movimentação, fornecendo um peso para cada uma delas, a partir da grandeza de saída do sistema *Fuzzy* denominada de QR ou qualidade da rota.

Assim, a cada iteração e para cada autômato, o sistema *Fuzzy* irá ser processado quatro vezes independentemente. Em cada um desses processamentos, ela irá levar em considerações as variáveis de entrada do sistema *Fuzzy* relativas a este autômato somente considerando uma das possíveis direções de movimentação.

Por exemplo, considerando que se está na primeira iteração e verificando o primeiro autômato. Inicialmente o sistema *Fuzzy* ira calcular a QR considerando, para esta iteração e este autômato, as grandezas de entrada calculadas para uma possível movimentação para direção esquerda, e gerará um QR_L (QR considerando a direção esquerda ou L-*left*). Depois ele irá, nesta mesma iteração e para este mesmo autômato, calcular a QR considerando, para esta iteração e este autômato, as grandezas de entrada calculadas agora para uma possível movimentação na direção acima, e gerará um QR_U (QR considerando a direção acima ou U-*up*).

De forma similar, o sistema *Fuzzy* irá calcular um QR_R (QR considerando a direção direita ou R-*right*) e um QR_D (QR considerando a direção abaixo ou D-*down*). O QR final resultante será o que resultar em seu maior valor e a rota a ser efetivamente seguida pelo autômato aquela correspondente a este QR escolhido. Assim:

$$QR_k = f_{Fuzzy}\,(\text{entradas}_k) \tag{1}$$

$$QR_{ke} = \max\,(QR_k) \tag{2}$$

$$ke = k \;\; \forall_{QR_{ke}=QR_k} \tag{3}$$

Onde

entradas = {DA; EP; RP; NE; EI; EA};

f_{Fuzzy} = função resposta característica do sistema *Fuzzy*;

k = opções de movimentação, ou seja: L (esquerda), U (acima), R (direita) e D (abaixo);

ke = a opção de movimentação efetivamente escolhida;

QR_k = cada um dos QR para cada uma da k opções de movimentação calculados pelo sistema *fuzzy*; e

QR_{ke} =QR da rota efetivamente escolhida.

Caso seja impossível de se fazer esta movimentação (caminho obstruído), será escolhido o próximo maior QR_k disponível (e a respectiva direção escolhida que o mesmo indica), e em caso de empate, a escolha será aleatória entre os de mesmo valor.

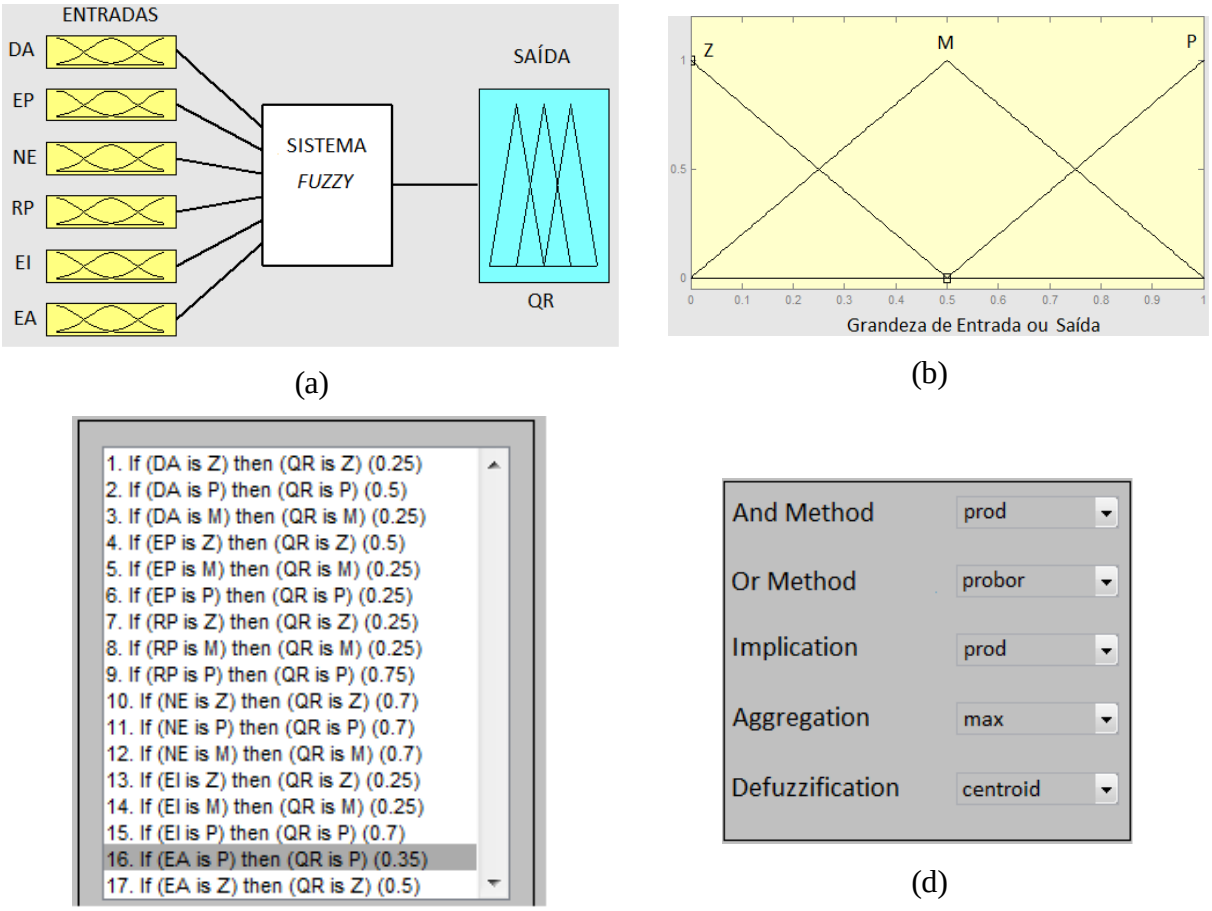
2.1.3 Sistema *fuzzy*

Todas estas grandezas de entrada são determinadas a cada iteração para cada autômato e para cada uma das opções possíveis de movimentação. Assim, não se tem, por exemplo, um único DA para um autômato em certa iteração, mas sim quatro DA´s ou um DA para cada opção de movimentação (esquerda, acima, direita e abaixo), calculados separadamente e que serão também utilizados em separado pelo sistema *fuzzy*, de modo a se poder valorar individualmente cada uma das opções de movimentação para cada um dos autômatos a cada iteração.

Como exemplos, tem-se na Figura 2 a visualização da etapa de inferência *Fuzzy*, com o processo de disparo das regras e a conversão *fuzzy*-escalar (*defuzzification*) do sistema considerando todas as grandezas de entrada como valendo 0,5. Na Figura 3 apresentam-se a indicação das grandezas envolvidas, das funções de pertinência, do conjunto de regras *fuzzy* com seus pesos relativos (ajustados empiricamente) e as normas e métodos para implicação, agregação e especificação utilizados.

Figura 2. Exemplos da etapa de inferência *fuzzy*, com o disparo das regras e especificação do valor de saída com todas as grandezas de entrada valendo 0,5

Fonte: adaptado de Braga et al. (2014).



(a)

(b)

(c)

(d)

Figura 3. Em (a) o sistema *fuzzy* com a indicação (em amarelo) das seis grandezas de entrada (DA, EP, NE, RP, EI e EA) e em azul da grandeza de saída (QR), em (b) a representação das funções de pertinência e termos primários para todas as grandezas, em (c) o conjunto de regras *fuzzy* com seus pesos relativos e em (d) as normas e métodos para implicação, agregação e especificação utilizados

Fonte: adaptado de Braga et al. (2014).

O sistema *fuzzy* possui então seis grandezas de entrada e uma grandeza de saída. Visando uma melhor padronização, todas as grandezas foram criadas como tendo o mesmo universo de discurso [0 a 1] e as mesmas três funções de pertinência no formato triangular com os seguintes termos primários: Z-zero; M-mediano e P-positivo. A cada iteração para cada autômato, o sistema *fuzzy* fará a leitura das seis grandezas de entrada e fornecerá a grandeza de saída independentemente para as quatro opções de movimentação (esquerda, acima, direita e abaixo), e então, como já mencionado, o algoritmo do programa selecionará aquela opção mais vantajosa.

2.2 Representação do conhecimento

Duas das mais primordiais formas de representar certo conhecimento são as funções e as tabelas. Uma possível vantagem de se representar o conhecimento de um sistema *fuzzy* específico em uma destas formas é na redução do tempo computacional para se obter um valor de saída. Espera-se que o tempo de processamento gasto no cálculo direto de uma única função, desde esta função seja relativamente pequena e simples, assim como o tempo de processamento necessário para se consultar uma tabela, especialmente se seu tamanho permitir que a mesma fique na memória dinâmica do computador, será muito menor do que o tempo de computacional requerido para se realizar todos os cálculos contidos numa etapa de inferência *fuzzy* completa.

Assim a ideia está em realizar previamente o levantamento de uma quantidade significativa de valores de entrada do sistema *fuzzy* e obter o respectivo valor de saída, perfazendo todas as etapas *fuzzy* necessárias pelo processo clássico. Então este conhecimento gerado deverá ser utilizado diretamente dentro da simulação multiagente, na forma de uma tabela que estará na memória dinâmica, ou através de uma única função, em substituição ao sistema *fuzzy* tradicional. Espera-se que a velocidade de processamento possa ser significativamente reduzida sem que haja perda na qualidade do resultado final. Esta função ou tabela substituta será denominada de função *fuzzy* ou tabela *fuzzy* respectivamente.

3 DESENVOLVIMENTO

3.1 Função *Fuzzy*

São inúmeras as ferramentas estatísticas existentes para a obtenção de uma função que represente certo conjunto de dados. Entretanto, pela inexistência de um padrão específico para um sistema *fuzzy* genérico, de início percebe-se que nenhuma única função estatística simples pode ser considerada como sendo a melhor opção de representação para todos os possíveis sistemas *fuzzy*. Isso acontece porque não existe nenhuma regra pré-definida sobre as funções de pertinência, sobre o conjunto de regras e seu pesos, assim como para as normas de implicação, agregação e especificação. Cada caso poderá ser único, e para cada situação particular terá que haver uma cuidadosa avaliação sobre a melhor forma de representação. Pode ser também que seja necessária mais de uma função (ou mesmo várias), pois um sistema *fuzzy* genérico pode possuir elementos e regras totalmente distintas dependendo dos valores de entrada.

Outro ponto que pesa contra as funções estatísticas são que os parâmetros normalmente

utilizados para se verificar o grau de atendimento de uma função podem não ser adequados. A Figura 4 ilustra esta situação. Nesta Cysne *et al.* (1998) correlacionaram os resultados de análises químicas obtidos para o teor de Ca encontrado em amostras de silício metálico (Simet) obtidos por dois métodos distintos de amostragem, um denominado de método padrão (MP) e outro denominado de método alternativo (MA). Caso ambos os métodos resultassem em amostras idênticas, e não existindo variação na amostragem e na análise das amostras, os pontos apresentados na Figura 4 estariam em perfeitamente em linha reta passando pela origem e com inclinação de 45º.

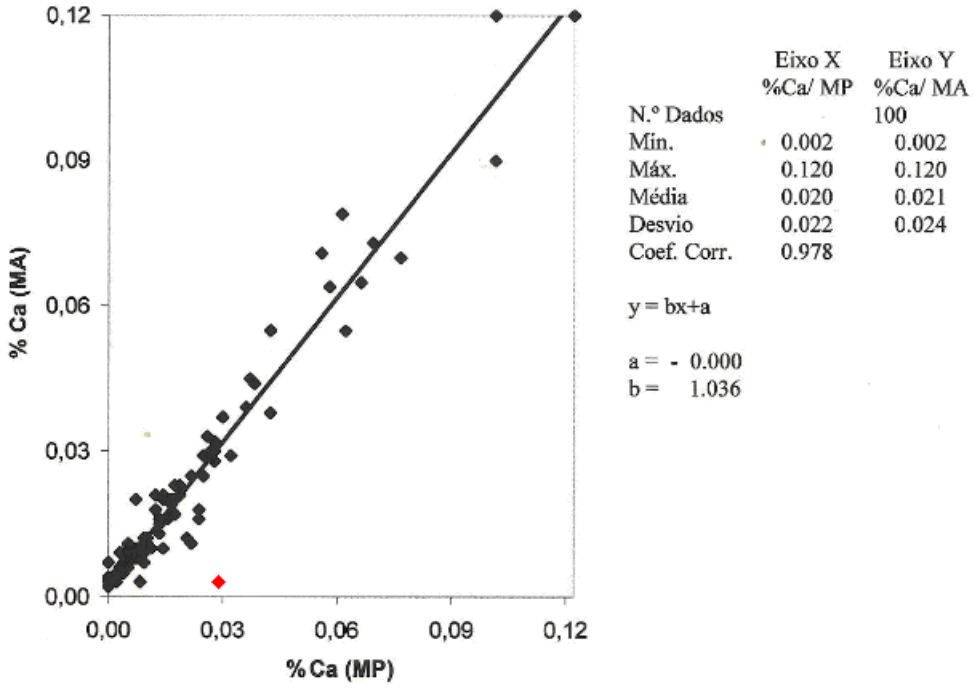


Figura 4. Correlação entre os valores de Ca em Simet obtidos por amostragem segundo o método padrão (MP) e o método alternativo (MA)

Fonte: adaptado de Cysne *et al.* (1998).

Visualmente pode-se considerar que a Figura 4 está aparentemente atendendo razoavelmente esta idealização, mas como se observaram algumas discrepâncias uma análise estatística foi realizada para melhor embasamento. Foi então concluído que os procedimentos padrão e alternativo podem ser considerados equivalentes (possuem mesma representatividade), pois o coeficiente de correlação encontrado entre ambos foi elevado (cerca de 0,98) para um grande número de amostras (n = 100) e através da determinação do intervalo de 95% de confiança (HOEL, 1992) dos coeficientes da reta de regressão indicarem que a reta de correlação pode passar pela origem e que sua inclinação pode ser de 45º.

Entretanto, o fato de serem estatisticamente equivalentes não impediu de haver divergências e desvios significativos entre muitos pontos. Na Figura 4 um ponto divergente foi destacado em vermelho. Supondo que o MP seria o equivalente ao uso do sistema *fuzzy* tradicional, e que o MA seria a função *fuzzy*, um ponto como este iria resultar em resultados muito distintos do esperado. No caso da simulação multiagente, uma pessoa que deveria ir para frente, por exemplo, poderia simplesmente começar a ir para trás, trazendo um comportamento anômalo ao sistema, ou causando um laço sem fim. Ou seja, não basta apenas

possuir um alto coeficiente de correlação entre a função *fuzzy* e o sistema *fuzzy*, a existência de desvios individuais significativos pode ter um papel preponderante.

Assim deste ponto em diante neste trabalho, apesar de ser uma possibilidade ainda em aberto e interessante para estudos posteriores, não será mais considerada a função *fuzzy*.

3.2 Dados *Fuzzy*

Uma alternativa simples seria no uso direto dos dados *fuzzy*, por exemplo, na forma de uma Tabela *Fuzzy* - TF. Idealmente esta TF deve ficar na memória dinâmica do computador, para redução no tempo de pesquisa, o que será uma grande vantagem. Uma questão é que as grandezas de entrada podem ter variações contínuas, ou seja, algum critério deverá ser adotado para a geração da TF.

Esta TF teria tantas colunas quantas fossem as grandezas de entrada e de saída. Assim, supondo por exemplo quatro grandezas de entrada e uma grandeza de saída, com uma combinação possível de 1.000 diferentes valores possíveis para as grandezas de entrada, a TF formada teria 5 colunas e 1.000 linhas, ou 5.000 campos. Uma alternativa para simplificar esta tabela e também facilitar enormemente a pesquisa sobre a mesma será utilizar o conceito de Matriz *Fuzzy*.

A Matriz *Fuzzy* - MF é uma matriz que possui somente os valores da(s) saída(s). Cada uma das grandezas de entrada é tratada como uma dimensão. Sabendo-se dos valores em cada dimensão (grandezas de entrada) se obtêm diretamente o valor da saída. Será necessário converter os valores de cada grandeza de entrada da TF no seu equivalente índice da matriz. A Figura 5 ilustra este processo de obtenção do valor de saída através de uma MF.

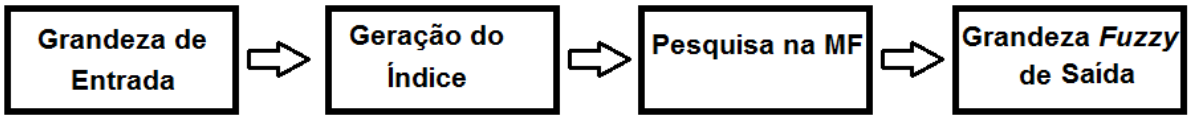


Figura 5. Processo de obtenção de um valor de saída através de uma MF

Alternativamente a MF, pode-se empregar uma linha ou Vetor *Fuzzy*- VF, onde todas as saídas estão ordenadas em uma única linha, mas de forma conhecida para que se possa facilmente converter as dimensões de entrada em um endereço nesta linha. Ressalta-se que o VF ou a MF são equivalentes, sendo apenas representações distintas dos mesmos dados. Neste trabalho vai-se adotar a MF.

4 RESULTADOS E DISCUSSÕES

Especificamente para o exemplo do programa Fuga, todas as grandezas de entrada (exceto NE) podem somente assumir os valores 0; 0,5 e 1, o que facilita especialmente a formação da TF. Para o NE se irá adotar uma variação de 0,1 entre (inclusive) os limites 0 e 1. Assim a TF a ser gerada terá 2.673 linhas (3 x 3 x 3 x 11 x 3 x 3 linhas) e sete colunas, sendo seis colunas com os valores de entrada (DA, EP, RP, NE, EI e EA) e uma coluna de saída (QR). A Tabela 1 representa parte desta TF gerada diretamente pelo Sistema *Fuzzy* (SF) do programa Fuga v. 1.0.

Tabela 1. TF parcial gerada pelo programa Fuga

DA	EP	RP	NE	EI	EA	QR
0	0	0	0	0	0	0,1633
0,5	0	0	0	0	0	0,3042
1	0	0	0	0	0	0,4439
0	0,5	0	0	0	0	0,3042
0,5	0,5	0	0	0	0	0,3042
1	0,5	0	0	0	0	0,4513
⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	0,5	1	1	1	1	0,6357
0,5	0,5	1	1	1	1	0,7028
1	0,5	1	1	1	1	0,7028
0	1	1	1	1	1	0,6683
0,5	1	1	1	1	1	0,7028
1	1	1	1	1	1	0,8367

Esta TF formada possui um total de 18.711 campos. Para simplificar esta representação se irá converter esta TF em sua respectiva MF. A Tabela 2 apresenta um exemplo de parte desta conversão para a grandeza de entrada DA. Deste modo se terá uma MF de seis dimensões (seis grandezas de entrada) e com apenas 2.673 campos totais. Isto implica em economia de memória já que os valores das grandezas de entrada são representados pelos próprios índices da MF. Outra vantagem da MF é que a pesquisa é facilitada, pois a busca sobre a MF se dá diretamente pelos valores dos índices. É verdade que a cada busca se terá que converter os valores da grandeza de entrada no seu equivalente índice, mas esta é uma operação computacionalmente muito mais simples de ser realizada que o conjunto de operações normalmente envolvidas num processamento *fuzzy* completo.

Tabela 2. Exemplo de conversão da TF para a MF

Tabela Fuzzy		**Matriz Fuzzy**	
Variável de Entrada	**Valor de Entrada**	**Respectiva Dimensão**	**Índice da Dimensão**
DA	0	DA_d	1
	0,5		2
	1		3

Para se verificar as diferenças entre o custo computacional por ambas as metodologias, foram realizados experimentos comparando o tempo de processamento para a realização de 10^6 cálculos completos do programa Fuga, tanto pelo SF tradicional como também pelo uso da respectiva MF equivalente. Este experimento foi repetido cinco vezes.

Os experimentos computacionais foram realizados em um computador ASUS Notebook, processador Intel Core i5 2.53 GHz, 64 bits, Windows 7, utilizando o programa Matlab 7.9.0 (R2009b) em seu ambiente próprio de programação e seu pacote *Toolbox Fuzzy* (FIS Editor). Os resultados dos tempos de processamento dos experimentos estão apresentados na Tabela 3.

Tabela 3. Tempos de processamento para o cálculo de 10^6 sistemas *fuzzy* completos pelo programa Fuga (valores das grandezas de entrada aleatórios)

Tempos de processamento (s)			$Ganho = \frac{tp - tb}{tm - tb}$
tb (tempo base)	tp (tempo Sistema *Fuzzy* convencional)	tm (tempo Matriz *Fuzzy*)	
0,6	1.637	12,3	139,9

Assim o tempo de processamento efetivo da obtenção do valor *fuzzy* de saída foi reduzido em cerca de 140 vezes, o que é muito expressivo e sem dúvida facilita a ampliação do campo de aplicação da lógica *fuzzy*. Em relação à qualidade do resultado, a única diferença esperada está associada ao arredondamento da grandeza NE para um número múltiplo de 0,1; já que as demais grandezas de entrada não sofreram alterações. Para visualizar o impacto deste arredondamento sobre o valor de saída (QR), a Figura 6 apresenta para um total de 10^6 conjuntos aleatórios de entrada (grandezas DA; EP; RP; NE; EI e EA) a variação entre os valores obtidos para QR tanto pelo SF do programa Fuga como pela sua equivalente MF.

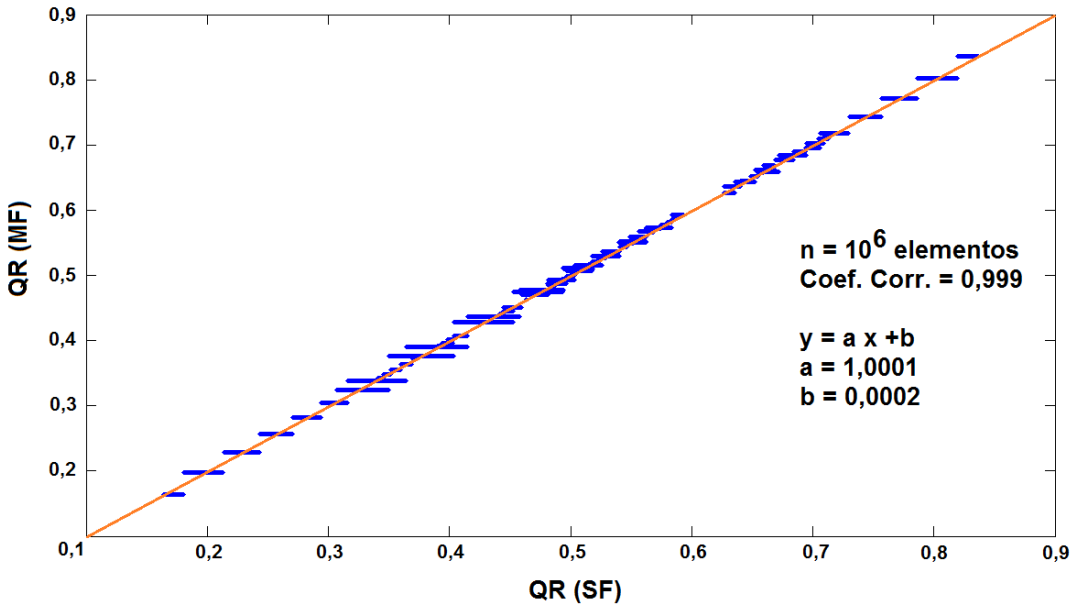


Figura 6. Correlação (em azul) entre os valores de QR obtidos pelo sistema *fuzzy* (SF) original e pela respectiva Matriz *Fuzzy* (MF)

Como se pode verificar o coeficiente de correlação encontrado foi muito elevado (≈ 1), sendo a melhor reta com coeficientes praticamente iguais ao da reta ideal, além de não terem sido verificados a ocorrência de qualquer único ponto significativamente fora da tendência.

Entretanto, a correlação apesar de excelente, não é perfeita devido aos arredondamentos no valor de NE. Para se evitar eventuais problemas, sugere-se que quando da efetiva implementação da lógica *fuzzy* por este método o ajuste empírico final seja feito sobre a MF e não sobre o SF.

5 CONSIDERAÇÕES FINAIS

É apresentado o conceito de Matriz *Fuzzy*, onde o conhecimento *fuzzy* de um sistema convencional é registrado na forma de uma matriz onde seus índices são as próprias grandezas de entrada do sistema *fuzzy*.

Em teste feito para o sistema *fuzzy* utilizado no programa Fuga v. 1.0, que realiza a simulação da movimentação humana em ambientes construídos, a utilização da Matriz *Fuzzy* como forma de representação do conhecimento *fuzzy* resultou em uma velocidade de processamento cerca de 140 vezes menor que a necessária para o processamento *fuzzy* convencional, sem perda significativa de precisão no resultado. Este é um resultado valoroso que contribui para que a lógica *fuzzy* possa ser usada mais eficientemente em simulações multiagentes de sistemas complexos.

Como sugestão para trabalhos futuros se tem a ampliação deste trabalho pela obtenção da Matriz *Fuzzy* e sua análise para uma maior gama de diferentes sistemas *fuzzy*. Outro ponto é que apesar da decisão da não implementação neste trabalho da função *fuzzy*, a possibilidade da viabilidade do seu uso continua em aberto para investigações futuras e maiores esclarecimentos.

AGRADECIMENTOS

Os autores agradecem a CAPES pelo suporte recebido.

REFERÊNCIAS

Almeida, P.E.M. & Evsukoff, A.G., 2003. Sistemas *Fuzzy*. In: REZENDE, S.O. (coord.). *Sistemas inteligentes*: Fundamentos e Aplicações, Cap. 7, Manole, p. 169-202.

Biondi Neto, L., Coelho, P.H.G., Amaral, J.L.M. & Mello, M.H.C.S., 2006. Minicurso de sistema especialista nebuloso. In: XXXVIII SBPO - Simpósio Brasileiro de Pesquisa Operacional, Goiânia. *Anais*..., Goiânia: Editora UGF, p. 2508-2543.

Boccara, N., 2004. *Modeling Complex System*, Springer.

Boulmakoul, A. & Mandar, M., 2011. Fuzzy Ant Colony for Virtual Pedestrian Simulation. *The Open Operational Research Journal*, v. 5, p. 19–29.

Braga, H.C., Moita, G.F., Camargo, F. & Almeida, P.E.M., 2014. Simulação da movimentação de pessoas em situações de emergência: aspectos ergonômicos e computacionais com autômatos Fuzzy e sua aplicação ao projeto arquitetônico. *Ambiente Construído*, v. 14, p. 61-77.

Cysne, M.A., Braga, H.C., Savini, G., Borges, I.T., Lamas, M.S. & Silveira, R.C., 1998. Otimização da metodologia para amostragem de silício metálico líquido refinado na Eletrosilex. In: XIII Seminário de Controle Químico em Metalurgia, Belo Horizonte. *Anais*…, São Paulo: ABM, 1998. v. 1. p. 219-232.

Hoel, P.G., 1992. Estatística Elementar, São Paulo: Atlas.

Marro, A.A., Souza, A.M.C., Cavalcante, E.R.S., Bezerra, G.S. & Nunes, R.O., 2010. *Lógica Fuzzy*: conceitos e aplicações. Universidade Federal do Rio Grande do Norte, 23 p.

Oliveira, H.A., Caldeira, A.M., Machado, M.A.S., Souza, R.C. & Tanscheit, R., 2007. *Inteligência Computacional Aplicada à Administração, Economia e Engenharia em Matlab*, São Paulo: Thompson.

Pan, X., Han, C.S., Dauber, K. & Law, K.H., 2007. A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations. *AI and Society*, n. 2, v. 22, p. 113-132.

Pelechano, N., Allbeck, J. & Badler, N., 2008. *Virtual crowds: methods, simulation, and control*, Morgan & Claypool Publishers.

Rojas, R., 1996. Fuzzy logic. In: *Neural networks*: a systematic introduction. Cap. 11, Berlim: Springer-Verlag, p. 297-318.

Simões, M.G. & Shaw, I.S., 2007. *Controle e Modelagem Fuzzy*, 2ª ed., São Paulo: Blücher: FAPESP.

Staal, M.A., 2004. *Stress, cognition and human performance*: a literature review and conceptual framework. NASA, 171 p. TM 212824.

Thompson, P.A. & Marchant, E.W., 1995. Testing and application of the computer model "SIMULEX". *Fire Safety Journal*, n. 24, p. 149-166.