

Integração Entre Lógica Nebulosa e Algoritmos Evolutivos

OMAR ANDRES CARMONA CORTES¹

Centro Federal de Educação Tecnológica do Maranhão - CEFET
Departamento Acadêmico de Informática - DAI
Av. Castelo Branco, 04 - Monte Castelo
São Luis - MA

¹omar.andres@terra.com.br

Resumo. O objetivo deste artigo é mostrar algumas das formas de integração entre a lógica nebulosa e os algoritmos evolutivos, sendo que a integração visa um aumento na qualidade da solução obtida. Após um panorama geral, mostra-se especificamente como aplicar os algoritmos evolutivos dentro da lógica nebulosa no refinamento de funções de pertinência. Nesse contexto, apresentam-se três tipos de refinamento: baseado em distâncias, baseado em intervalos de desempenho e baseados em centro-largura. Uma aplicação de cada tipo de refinamento é mostrada em estudos de caso. Dentre os algoritmos evolutivos considerados, aborda-se a utilização de algoritmos genéticos e estratégias evolutivas. Uma discussão sobre os métodos mostrados também é apresentada.

Palavras-Chave: Lógica Nebulosa, Algoritmos Evolutivos, Algoritmos Genéticos, Estratégias Evolutivas, Otimização.

1 Introdução

A inteligência artificial (IA) tem despertado bastante interesse em diversas comunidades. Basicamente o objetivo da inteligência artificial é adicionar a capacidade de raciocínio a um computador. Dentre as diversas técnicas de IA existentes estão a lógica nebulosa (LN) [11] e os algoritmos evolutivos (AEs) [13].

A lógica nebulosa é uma extensão da lógica tradicional, onde os valores 0 e 1 (verdadeiro e falso) da lógica tradicional são estendidos para valores contínuos nesse mesmo intervalo. Essa característica da lógica nebulosa permite que computadores possam trabalhar com incerteza e imprecisão (características intrínsecas do conhecimento humano).

A teoria dos algoritmos evolutivos baseia-se na teoria da evolução de Darwin, onde os indivíduos mais fortes interagem entre si propagando seu genes e gerando descendentes mais aptos às novas condições.

A integração entre algoritmos evolutivos e a lógica nebulosa pode ocorrer de diversas formas e tem como principal objetivo um aumento na qualidade de uma solução. Na LN a qualidade da solução pode ser, por exemplo, uma maior aproximação do resultado fornecido pelo sistema nebuloso ¹ do resultado esperado. Enquanto que no caso de AEs, um aumento na qualidade da solução pode ser, por exemplo, um resultado ótimo

¹Qualquer sistema que utilize a lógica nebulosa como técnica de IA é conhecido como sistema nebuloso.

mais próximo do mínimo/máximo global.

Neste trabalho destaca-se a utilização de AEs no refinamento de funções de pertinência de sistemas nebulosos. Para cumprir seu objetivo este artigo está dividido da seguinte maneira: as Seções 2 e 3 apresentam os conceitos básicos da lógica nebulosa e dos algoritmos evolutivos, respectivamente; a Seção 4 descreve as diversas formas como a LN e os algoritmos genéticos (AG) podem ser integrados; na Seção 5 são discutidos alguns estudos de caso; finalmente, na Seção 7 são apresentadas as conclusões deste trabalho.

2 Lógica Nebulosa

A principal diferença entre a lógica tradicional e a LN é a faixa de valores considerados como verdadeiro. Enquanto que na lógica tradicional existem somente os valores 0 e 1 (verdadeiro e falso), na LN os valores verdadeiro e falso são expressos em termos de graus.

Basicamente um sistema nebuloso funciona como mostrado na Figura 1. O sistema recebe uma entrada que é *fuzzificada* utilizando funções de pertinência. Em seguida, o valor fuzzificado é passado por uma máquina de inferência. Finalizando, o resultado da máquina de inferência é transformado em um valor real através de um método de *desfuzzificação* [11, 17]. A base de conhecimento é formada por uma base de dados (funções de pertinência) e por uma base de regras.

A Figura 2 mostra um exemplo de função de perti-

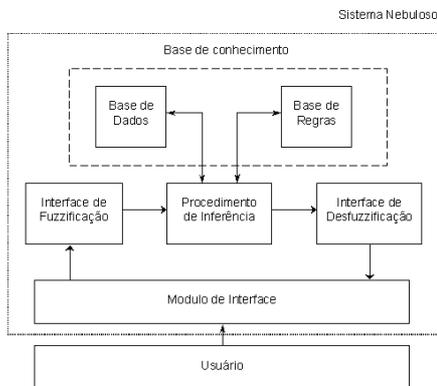


Figura 1: Sistema Nebuloso

nência, o qual atribui um valor qualitativo à temperatura da água, sendo que temperatura passa a ser uma variável nebulosa. A função de pertinência apresentada esta dividida em três conjuntos nebulosos, sendo que dada uma temperatura qualquer, esta pode pertencer a um ou mais conjuntos ao mesmo tempo, o que não pode ocorrer na lógica tradicional. Por exemplo, a temperatura 30° pertence ao conjunto fria com um grau de aproximadamente 0,2 e morna com um grau de aproximadamente 0,8.

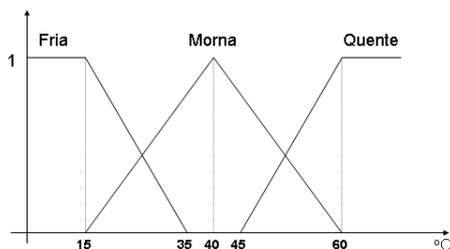


Figura 2: Exemplo de Função de Pertinência

O processo de inferência é baseado em regras do tipo **SE Antecedente THEN Conseqüente**. Essas regras são armazenadas na base de conhecimento. Uma regra é um relacionamento entre dois ou mais conjuntos nebulosos, por exemplo: SE $X = A$ ENTÃO $Y = B$, onde A e B são termos lingüísticos, e X e Y são variáveis lingüísticas em um certo domínio. Um exemplo de uma regra real é apresentado na Equação 1.

$$\text{SE pressão} = \text{baixa ENTÃO valvula} = \text{aberta.} \quad (1)$$

Basicamente o processo de inferência é responsável pelo raciocínio de um sistema nebuloso. Como ocorre

matematicamente o processo de raciocínio pode ser visto em [11, 16]

3 Algoritmos Evolutivos

Basicamente, um algoritmo evolutivo processa uma população de possíveis candidatos a uma solução ótima. Cada indivíduo ou cromossomo é formado por genes, onde cada gene reflete um parâmetro ou variável de um sistema.

Um AE funciona da seguinte maneira: (1) Inicializa-se uma população de possíveis soluções; (2) Avalia-se a população gerada; (3) Selecionam-se os pais para gerar uma nova população; (4) Os pais são recombinados gerando uma nova população; (5) A nova população sofre mutação; (6) A nova população substitui os genitores ou compete com os pais pela sobrevivência de acordo com o algoritmo adotado; (7) Se o critério de término for atingindo o algoritmo é finalizado, caso contrário retorna para o passo (2).

Algoritmo 1 - Estrutura de um Algoritmo Evolutivo

```

t ← 0
inicializar P(t)
Avaliar P(t)
enquanto Critério de Parada não Atingido faça
    t ← t + 1
    Seleciona P(t) de P(t - 1)
    Altera P(t)
    Avalia P(t)
fim enquanto

```

Cada iteração do processo é chamada de geração. Geralmente o critério de parada é uma certa quantidade de gerações, embora outros critérios, como a comparação de erros, possam ser utilizados.

Para avaliar a capacidade de um indivíduo como uma possível solução é necessário utilizar uma função objetivo, a qual deve fornecer um valor qualitativo da possível solução, esse valor qualitativo, chamado de *Aptidão*, vai determinar o quão forte é esse indivíduo dentro da população. Os indivíduos mais fortes tem melhores chances de propagar seus genes para um próxima geração. Nesse sentido, operadores genéticos - *cruzamento* e *mutação* - são utilizados para gerar os novos descendentes. Dessa forma, a qualidade das soluções aumenta a medida que novos ciclos de cruzamento e mutação são gerados [13].

O objetivo do cruzamento ou recombinação é compartilhar informações entre indivíduos. Com ele combinam-se as capacidades de dois pais para formar um ou mais descendentes, com a possibilidade de gerar um indivíduo que futuramente irá gerar indivíduos ainda mais

fortes [9]. Já o objetivo da mutação é alterar um ou mais genes para impedir uma convergência prematura da solução.

Os algoritmos genéticos (AGs) e as estratégias evolutivas (EEs) compartilham as mesmas características gerais tais como: uma população de indivíduos, um esquema de seleção, uma política de substituição e ambos utilizam algum tipo operador genético. Entretanto, dentro das características gerais apresentadas existem algumas diferenças:

- **Representação de Indivíduos:** Os indivíduos podem ser representados de diversas formas. A diferença é que cada indivíduo nos AGs é representado por apenas um vetor de reais ². Enquanto que nas EEs um indivíduo é representado por um par de vetores na forma (x, σ) . O primeiro vetor de objetos (x_1, \dots, x_n) constitui um candidato à solução dentro do espaço de busca. O segundo vetor $(\sigma_1, \dots, \sigma_n)$ representa o desvio padrão que será aplicado a cada elemento x_i quando o mesmo sofrer mutação. No caso das EEs os operadores genéticos são aplicados ao par de vetores.
- **Seleção de Indivíduos e Aplicação dos Operadores Genéticos:** Nos AGs os indivíduos são primeiro selecionados e em seguida utilizam-se os operadores genéticos para gerar novos indivíduos. Enquanto que nas EEs a seleção é feita depois da utilização dos operadores genéticos.
- **Política de Substituição:** De maneira geral, nos AGs os novos indivíduos gerados substituem os pais. Nas EEs os indivíduos são selecionados baseando-se na sua *Aptidão*, ou seja, os melhores indivíduos passam para a próxima geração. Nesse contexto, os indivíduos nas EEs são selecionados apenas entre os recém gerados $EE - (\mu, \lambda)$ ou competem com seus pais pela sobrevivência $EE - (\mu + \lambda)$, onde μ representa a quantidade de pais e λ representa a quantidade de descendentes.

4 Integração entre LN e AEs

Os algoritmos evolutivos podem ser utilizados no desenvolvimento de sistemas nebulosos desde a aquisição de conhecimento até o refinamento do sistema. A aquisição de conhecimento é uma das tarefas mais complexas da modelagem de sistemas nebuloso, pois os especialistas muitas vezes não tem idéia de como expres-

²Quando um AG é representado por um vetor de reais ele é chamado de *Real-Coded*. No caso de ser representado por um vetor de binários ele é dito *binary-coded*.

sar o próprio conhecimento ³. Nesse contexto, os algoritmos evolutivos podem ser utilizado em três etapas diferentes do processo de desenvolvimento de sistemas nebulosos:

- **Geração automática de regras** - as regras que compõem a base de regras podem ser geradas tendo como base exemplos pré-definidos.
- **Simplificação de regras** - É possível que no processo de geração das regras haja uma sobrecarga de aprendizado. Este comportamento ocorre quando os exemplos são satisfeitos com um grau maior que o desejado através de regras redundantes. Para solucionar esse problema e melhorar a precisão do sistema nebuloso é necessário retirar as regras redundantes.
- **Ajuste de funções de pertinência e das matrizes de conhecimento** - Nesta etapa pode-se ajustar os parâmetros da base de dados, ou seja, ajustar as funções de pertinência. Para utilizar esta etapa separadamente pressupõem-se que já exista um sistema nebuloso.

Especificamente, neste trabalho têm-se o interesse no refinamento das funções de pertinência, pois a teoria da interação entre LN e AEs é relativamente vasta. A aplicação dos AEs pode-se estender também aos demais tipos de algoritmos evolutivos. A Figura 3 mostra um modelo para um sistema nebuloso genético. Basicamente, em cada geração o AG altera os dados contidos na base de conhecimento enquanto o sistema nebuloso fornece as novas saídas. No término da execução o AG deve ser capaz de apresentar a melhor de todas as soluções avaliadas.

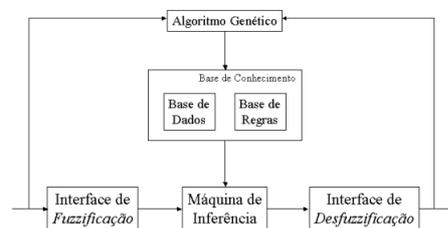


Figura 3: Modelo de um Sistema Nebuloso Genético

4.1 Tipos de Refinamento em Funções de Pertinência

Nesta seção são mostradas três formas de realizar o refinamento de funções de pertinência: refinamento baseado

³Em um sistema nebuloso, o conhecimento de um especialista é armazenado na base de regras.

em distâncias [10], refinamento baseado em intervalos de desempenho [15] e refinamento baseado no centro-largura [5].

4.1.1 Refinamento baseado em distâncias

O refinamento baseado em distância segue o esquema da Figura 4. Os extremos de cada conjunto nebuloso, a partir do segundo conjunto até o penúltimo, devem necessariamente coincidir com o centro dos conjuntos nebulosos vizinhos. Para cada conjunto nebuloso o centro c_i de A_i com índice $i < j$ deve permanecer a esquerda do conjunto nebuloso A_j respeitando $c_i < c_j$. Essa propriedade permite representar a função de pertinência apenas utilizando as distâncias Δ_i entre conjuntos adjacentes, sendo que os centros são obtidos pela Equação 2.

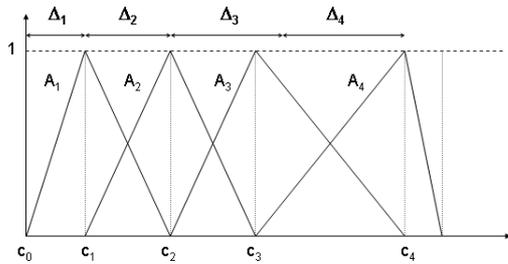


Figura 4: Refinamento baseado em distâncias

$$c_i = c_{i-1} + \Delta_i = c_0 + \sum_{k=1}^i \Delta_k \quad (2)$$

Cada indivíduo é representado por um vetor de reais, onde cada elemento do vetor corresponde a um Δ_i , ou seja, existe um único parâmetro para cada conjunto nebuloso A_i . Supondo uma função de pertinência com 5 conjuntos nebulosos, sua representação cromossômica seria: $(\Delta_1, \Delta_2, \Delta_3, \Delta_4, \Delta_5)$.

4.1.2 Refinamento baseado em intervalos de desempenho

No refinamento baseado em intervalos de desempenho, o conjunto nebuloso pode ser alterado conforme o intervalo de desempenho selecionado, como pode ser visto na Figura 5. Os intervalos de desempenho são definidos através das Equações 3, 4 e 5.

$$[a_t^l, a_t^r] = [a_t - \frac{b_t - a_t}{2}, a_t + \frac{b_t - a_t}{2}] \quad (3)$$

$$[b_t^l, b_t^r] = [b_t - \frac{b_t - a_t}{2}, b_t + \frac{c_t - b_t}{2}] \quad (4)$$



Figura 5: Refinamento baseado em intervalos de desempenho

$$[c_t^l, c_t^r] = [c_t - \frac{c_t - b_t}{2}, a_c + \frac{c_t - b_t}{2}] \quad (5)$$

Cada indivíduo é representado por todos os pontos que formam a função de pertinência. Dessa forma, a representação da Figura 6 seria na forma de:

$$I_i = (a_1, b_1, c_1, a_2, b_2, c_2) \quad (6)$$

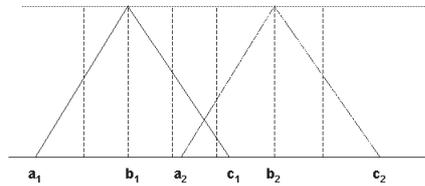


Figura 6: Refinamento baseado em intervalos de desempenho - Exemplo

A equação 6 representa o que seria o primeiro indivíduo da população. Os demais indivíduos são gerados aleatoriamente nos respectivos intervalos de desempenho.

4.1.3 Refinamento baseado no centro-largura

Este tipo de refinamento é específico para funções de pertinência do tipo gaussiana, as quais são formadas pela Equação 7, onde σ representa a largura da curva e c representa o centro da curva. Nesse contexto, no refinamento podem ser alterados o centro e a largura da curva como mostrado na Figura 7.

$$e^{-\frac{(x-c)^2}{2*\sigma^2}} \quad (7)$$

Outro ponto que deve ser destacado é que em um sistema nebuloso, soluções fora da região de busca podem comprometer seriamente o comportamento do sistema invertendo a posição correta dos conjuntos nebu-

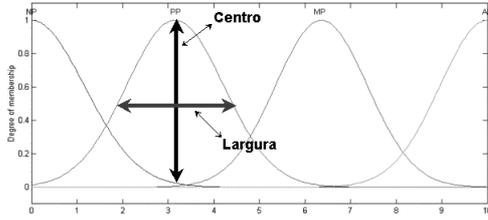


Figura 7: Refinamento baseado no centro-largura

losos. Dessa forma, no refinamento baseado no centro-largura as seguintes restrições devem ser obedecidas:

- **Não variar o centro dos conjuntos nebulosos que estão nas extremidades** - uma variação no centro desses conjuntos poderia gerar valores fora do universo de discurso de uma variável nebulosa.
- **Entre dois conjuntos subseqüentes, o primeiro conjunto deve ter seu centro sempre menor que o segundo** - isso evita que o significado dos conjuntos nebulosos seja invertido alterando totalmente o comportamento do sistema nebuloso.
- **Variação da largura das curvas entre 0.7 e 2.0** - valores inferiores a 0.7 geram conjuntos gaussianos muito estreitos e valores superiores a 2.0 geram conjuntos muito largos.

Neste tipo de refinamento, os indivíduos gerados fora da especificação são punidos com Aptidão = 0, ou seja, os indivíduos perdem a chance de se reproduzir pois, não chegam a ser avaliados. [5]

5 Aplicações

Para mostrar a efetividade dos métodos de refinamento em funções de pertinência, nesta seção são mostrados três estudos de caso. Um estudo de caso para cada tipo de refinamento apresentado.

5.1 Estudo de Caso 1

No trabalho de Frank Hoffman [10] utiliza-se o refinamento baseado em distância no desenvolvimento de um sistema nebuloso de controle de robôs. O objetivo do refinamento é encontrar um ponto de equilíbrio entre a quantidade de navegação necessária e ao mesmo tempo manter uma distância segura dos obstáculos.

O AE utilizado é uma estratégia evolutiva com 30 indivíduos. O critério de parada é uma quantidade máxima de colisões ou uma determinada quantidade de tempo, sendo que a média apresentada é de 50 gerações. Quando ocorre uma colisão a aptidão do indivíduo é

penalizada. Em outras palavras, quanto menos colisões o robô tiver mais forte será o indivíduo.

No experimento realizado o robô tem a tendência a virar para a direita para que o mesmo consiga sair caso fique preso em um canto. Quando o robô é apresentado a um ambiente desconhecido o comportamento e o desempenho se mostraram semelhantes ao do ambiente já conhecido.

5.2 Estudo de Caso 2

Neste estudo de caso, Cordon [3] aplica o refinamento baseado em intervalos de desempenho para aproximar a superfície de saída (em R^3) das funções matemáticas mostradas nas Equações 8, 9 e 10, respectivamente.

$$F_1(x_1, x_2) = x_1^2 + x_2^2 \quad (8)$$

$x_1, x_2 \in [-5, 5]$ e $F_1(x_1, x_2) \in [0, 50]$

$$F_2(x_1, x_2) = x_1^2 + x_2^2 - \cos 18x_1 - \cos 18x_2 \quad (9)$$

$x_1, x_2 \in [-1, 1]$ e $F_2(x_1, x_2) \in [2, 3.5131]$

$$F_3(x_1, x_2) = 10 \frac{x_1(1-x_2)}{x_1(1-2x_2)+x_2} \quad (10)$$

$x_1, x_2 \in [0, 1]$ e $F_3(x_1, x_2) \in [0, 10]$

O algoritmo evolutivo utilizado é um algoritmo genético com 500 e 1000 gerações com 61 indivíduos. Sendo que o sistema nebuloso após ajustado apresenta um erro em torno de no máximo 5%.

5.3 Estudo de Caso 3

O objetivo do trabalho de Cortes [6] é mostrar um sistema nebuloso desenvolvido para determinar o grau de portabilidade de benchmarks paralelos. Com esse grau em mãos o usuário é capaz de decidir se o benchmark pode comparar diferentes arquiteturas de maneira confiável.

Foram utilizados tanto algoritmos genéticos quanto estratégias evolutivas com 70 e 110 indivíduos. Na execução dos experimentos foram consideradas 100 e 500 gerações. O tipo de refinamento é o baseado no centro-largura, sendo que está se levando em conta a precisão do sistema nebuloso antes e após o refinamento, como mostrado na Tabela 1. Neste caso, observa-se que para a aplicação considerada os algoritmos genéticos tiveram um desempenho um pouco melhor. Entretanto essa diferença não é considerável se feito um teste de hipótese com nível de significância de 1%. Nesse contexto, pode-se dizer que ambos os algoritmos evolutivos são adequados para o refinamento deste tipo de aplicação.

Tabela 1: Precisão do Sistema

Benchmark	Sistema Nebuloso	Após AG	Após EE
Precisão	75.0%	99.80%	99.40%

6 Discussão

O processo de ajuste automático de funções de pertinência pode diminuir consideravelmente o tempo necessário para a finalização de um sistema nebuloso, pois o ajuste manual pode exigir um conhecimento profundo do universo de discurso cujas funções de pertinência estão sendo refinadas.

Dentre as formas de representação das funções de pertinência, a que gera indivíduos com menos genes, ou seja, utiliza vetores menores é a do refinamento baseado em distância. Conseqüentemente, o tempo de processamento para concluir o ajuste pode ser bem menor do que das demais técnicas de refinamento. Isso dependerá também da quantidade de conjuntos nebulosos das funções de pertinência que estão sendo ajustadas.

Nos AE considerados, sugere-se a representação dos indivíduos apenas com vetores de números reais, pois a representação na forma binária em AGs aumenta o tamanho do vetor utilizado. Além disso, a precisão da solução é afetada em AGs *binary-coded* a medida que aumenta a quantidade de conjuntos nebulosos da função de pertinência.

No estudo de caso 3, os indivíduos gerados fora da especificação são penalizados com (Aptidão = 0), ou seja, não serão avaliados. Esse comportamento tem um ponto positivo com relação ao desempenho (tempo de execução), já que avaliar um indivíduo é um processo computacionalmente pesado⁴. Nesse contexto, pode haver uma diminuição considerável no tempo de execução do AE.

A quantidade de gerações consideradas para o refinamento é muito dependente da aplicação. Nos três estudos de caso mostrados existe uma variação considerável na quantidade de gerações. Entretanto, todos os estudos apresentaram melhorias significativas.

7 Conclusões

Neste artigo apresentou-se a teoria de como realizar o refinamento de funções de pertinência de sistemas nebulosos. Em especial, mostrou-se o refinamento de funções

⁴Cada indivíduo representa um sistema nebuloso. Nesse contexto, para avalia-lo é necessário construir o sistema nebuloso tendo como base os seus genes; fornecer os dados de entrada; comparar os dados fornecidos pelo sistema nebuloso com os dados esperados; e, calcular a aptidão do indivíduo.

de pertinência de sistemas nebulosos.

Nos três estudos de caso mostrados, o refinamento aumentou o desempenho dos sistemas nebulosos. Em outras palavras, o erro produzido pelos sistemas nebuloso diminuiu consideravelmente após o refinamento.

8 Trabalhos Correlatos

Neste trabalho discutiu-se a utilização de AEs na LN. Entretanto, a lógica nebulosa pode também ser utilizada dentro de algoritmos evolutivos. Nesse contexto, esta seção apresenta outras aplicações da interação entre a lógica nebulosa e os algoritmos evolutivos.

No trabalho de Hoffman [10], propõem-se também a utilização de estratégias evolutivas com os seguintes objetivos: mapear o universo de discurso de uma variável em valores normalizados; e obter um método automático de aprendizagem de regras. Nessa linha de obtenção automática de conhecimento estão os trabalhos de Castillo [2], Cordón [4] e Cassillas [1].

Oliveira [14], utiliza lógica nebulosa para controlar parâmetros de evolução de AGs. Os AGs resultantes são aplicados na resolução de problemas *NP-Hard*, como por exemplo o problema do caixeiro viajante. Enquanto que Liu [12] explora os algoritmos genéticos na otimização de funções multimodais. Neste último exemplo, a junção com a lógica nebulosa ocorre na representação dos indivíduos que neste caso são vetores nebulosos.

No trabalho de Herrera [9] é feita uma análise de diversos mecanismos de cruzamento e de mutação em AGs. Dentre os mecanismos de cruzamento analisados encontra-se um mecanismo baseado em conectivas nebulosas [7, 8].

Referências

- [1] J. Casillas, O. Cordón, M. J. del Jesus, and F. Herrera. Genetic Tuning of Fuzzy Rule Deep Structures for Linguistic Modeling. Technical report, Department of Computer Science - University of Granada, Janeiro 2001. DECSAI-010102.
- [2] L. Castillo, A. González, and R. Pérez. Including a Simplicity Criterion in The Selection of the Best Rule in a Genetic Fuzzy Learning Algorithm. *Fuzzy Sets and Systems*, 120:309–321, 2001.
- [3] O. Cordón and F. Herrera. A Three-Stage Evolutionary Process for Learning Descriptive and Approximative Fuzzy-Logic-Croled Knowledge Bases From Examples. *International Journal of Approximate Reasoning*, 17:369–407, 1997.

- [4] O. Cordón and F. Herrera. Hybridizing Genetic Algorithms With Sharing Scheme and Evolution Strategies for Designing Aproximative Rule-based system. *Fuzzy Sets and Systems*, 118:235–255, 2001.
- [5] O. A. C. Cortes. *Um Sistema Nebuloso Evolutivo para Determinar o Grau de Portabilidade de Benchmarks Paralelos*. PhD thesis, Instituto de Ciências Matemáticas e de Computação - USP, São Carlos - SP, Março 2004.
- [6] O. A. C. Cortes, R. H. C. Santana, and M. J. Santana. Determinação do Grau de Portabilidade de Benchmarks Paralelos Através de Lógica Nebulosa. *Revista de Informática Teórica e Aplicada*, 9(3):000–000, 2002.
- [7] F. Herrera, M. Lozano, and J. L. Verdagay. Fuzzy Tools to Improve Genetic Algorithms. In *Second European Congress on Intelligent Techniques and Soft Computing*, volume 3, pages 1532–1539, Aachen, Alemanha, 1994.
- [8] F. Herrera, M. Lozano, and J. L. Verdagay. The Use of Fuzzy Connectives to Design Real-Coded Genetic Algorithms. *Mathware and Soft Computing*, 1(3):239–251, 1995.
- [9] F. Herrera, M. Lozano, and J. L. Verdagay. Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis. *Artificial Intelligence Review*, 4(12):265–319, 1998.
- [10] F. Hoffmann. Evolutionary algorithms for fuzzy control system design. In *Proceedings of the IEEE*, volume 89, pages 1318–33, September 2001.
- [11] G. J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, New Jersey, 1995.
- [12] B. Liu and K. Iwamura. Fuzzy Programming With Fuzzy Decision and Fuzzy Simulation-Based Genetic Algorithms. *Fuzzy Sets and Systems*, 122:253–262, 2001.
- [13] Z. Michalewicz. *Genetic Algorithms + Data Structure = Evolution Programs*. Springer-Verlag, New York, 3 edition, 1999.
- [14] A. Oliveira and E. Nascimento. Uma Contribuição ao Estudo de Algoritmos Evolutivos Fuzzy. In *Proc VI Iberamia*, pages 175–186, Lisboa, 1998.
- [15] L. Renhou and Z. Yi. Fuzzy Logic Controller Based on Genetic Algorithms. *Fuzzy Sets and Systems*, 83:1–10, 1996.
- [16] S. Sandri and C. Correa. *Lógica Nebulosa*. Instituto de Pesquisas Espaciais (INPE), 1999.
- [17] I. S. Shaw and M. G. Simões. *Controle e Modelagem Fuzzy*. Edgard Blücher LTDA, 1999.