Universidade do Estado do Rio de Janeiro
Centro de Tecnologia e Ciências
Faculdade de Engenharia

Ana Carolina Xavier Silva Fonseca

Modelo Evolucionário Baseado em Agregador Fuzzy para Avaliação de Múltiplos Objetivos

Rio de Janeiro
2017

Ana Carolina Xavier Silva Fonseca

Modelo evolucionário baseado em agregador fuzzy para avaliação de múltiplos objetivos

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes e Automação.

Orientador: Prof. Dr. José Franco Machado do Amaral

Rio de Janeiro

2017

CATALOGAÇÃO NA FONTE

UERJ / REDE SIRIUS / BIBLIOTECA CTC/B

F676 Fonseca, Ana Carolina Xavier Silva.
Modelo evolucionário baseado em agregador fuzzy para avaliação de múltiplos objetivos / Ana Carolina Xavier Silva Fonseca. – 2017.
142f.

Orientador: José Franco Machado do Amaral.
Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia.

1. Engenharia Eletrônica. 2. Sistemas difusos - Dissertações. 3. Algoritmos genéticos - Dissertações. 4. Otimização multiobjetivo - Dissertações. I. Amaral, José Franco Machado do. II. Universidade do Estado do Rio de Janeiro. III. Título.

CDU 007.52

_____ _____

Assinatura Data

Ana Carolina Xavier Silva Fonseca

Modelo evolucionário baseado em agregador fuzzy para avaliação de múltiplos objetivos

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes e Automação.

Aprovado em 22 de fevereiro de 2017.

Banca Examinadora:

_____

Prof. Dr. José Franco Machado do Amaral (Orientador)

Faculdade de Engenharia – UERJ

_____

Prof. Dr. Pedro Henrique Gouvêa Coelho

Faculdade de Engenharia – UERJ

_____

Profª. Dra. Marley Maria Bernardes Rebuzzi Vellasco

Departamento de Engenharia Elétrica – PUC-RJ

Rio de Janeiro

2017

DEDICATÓRIA

Aos meus pais.

AGRADECIMENTOS

Agradeço especialmente aos meus pais pela educação que me proporcionaram, por todo amor, apoio e carinho.

Ao meu orientador, Prof. José Franco Machado do Amaral, pela orientação necessária para o desenvolvimento deste trabalho ao longo desses anos.

Aos professores do Programa de Pós-Graduação em Engenharia Eletrônica por compartilharem seus conhecimentos.

Aos amigos do mestrado pelo companheirismo e apoio.

Aos membros da banca examinadora pela disponibilidade e contribuição.

À UERJ e ao PEL que me proporcionaram a oportunidade de realizar o mestrado.

À CAPES pelo apoio financeiro.

A todos aqueles que contribuíram direta e indiretamente durante o período de execução deste trabalho.

Nossa maior fraqueza está em desistir.

O caminho mais certo de vencer é sempre tentar mais uma vez.

Thomas Edison

RESUMO

FONSECA, Ana Carolina Xavier Silva. *Modelo evolucionário baseado em agregador fuzzy para avaliação de múltiplos objetivos.* 2017. 142 f. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2017.

Este trabalho apresenta um modelo evolucionário para viabilizar a evolução de sistemas inteligentes, baseado em um algoritmo genético que utiliza a lógica fuzzy no processo de avaliação de múltiplos objetivos. O método de avaliação tradicional dos algoritmos genéticos é modificado, de forma que um sistema fuzzy é executado durante o processo de avaliação, agregando os diversos objetivos do problema em questão e gerando um valor de aptidão para cada indivíduo. O modelo proposto apresenta uma forma mais simples e interpretável de inserir preferências e/ou especificações, pois utiliza a lógica fuzzy para isso. Tais preferências são inseridas antes do processo de evolução, garantindo que a evolução seja guiada na direção desejada, evitando a necessidade de uma intervenção do projetista ao final do processo para escolha da solução mais adequada. Além da implementação no ambiente puramente simulado, o modelo ainda prevê uma interface para aplicações reais utilizando um hardware microcontrolado, possibilitando um ambiente real de aquisição de dados de entrada e saída para futuras aplicações. Estudos de casos em áreas de aplicações distintas são avaliados através de simulações computacionais e comparados a resultados encontrados por outras técnicas. Entre os estudos de casos, são incluídos problemas *benchmark* para avaliação multiobjetivo, projeto de sistemas fuzzy, projeto de controladores PID e projeto de controladores fuzzy. Um outro estudo de caso foi também desenvolvido para testar a funcionalidade de um protótipo simples do hardware microcontrolado. As ferramentas desenvolvidas para evolução, avaliação e implementação apresentaram bom desempenho nos estudos de casos analisados, podendo ser utilizadas como base para novas aplicações e implementações de sistemas reais.

Palavras-chave: Sistemas fuzzy; Algoritmos genéticos; Otimização multiobjetivo; Sistemas híbridos.

ABSTRACT

FONSECA, Ana Carolina Xavier Silva. *Evolutionary model based on fuzzy aggregator for multiobjective evaluation.* 2017. 142 f. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2017.

This work presents an evolutionary model to enable the evolution of intelligent systems, based on a genetic algorithm and using the fuzzy logic in the evaluation process of multiple objectives. The traditional evaluation method of genetic algorithms is modified, thus a fuzzy system is executed during the evaluation process, aggregating the many objectives of the problem and generating a fitness for each individual. The proposed model presents a simplest and more interpretable way to insert preferences and/or specifications because it uses the fuzzy logic. These preferences are introduced before the evolution process ensuring that the evolution is guided in the desired direction, avoiding the necessity of a designer intervention at the final of the process to choose the most appropriate solution. Further the implementation in the purely simulated environment, the model also has a platform with an interface to real applications using a microcontrolled hardware, making possible a real environment for acquisition of input and output data for future applications. Case studies in different application areas are analyzed through computational simulations and compared to results found by other techniques. The case studies include benchmark problems for multiobjective evaluation, fuzzy system design, PID control design and fuzzy control design. Another case study was developed to test the functionality of a simple microcontrolled hardware prototype. The tools developed for evolution, evaluation and implementation presented good performance in the case studies analyzed and can be used as a basis for new applications and implementation of real systems.

Keywords: Fuzzy systems; Genetic algorithms; Multiobjective optimization; Hybrid systems.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AG	Algoritmo Genético
A/D	Analógico / Digital
BR	Base de Regras
CC	Corrente contínua
FD	*Fuzzy Pareto Dominance*
FP	Função de Pertinência
GAOT	*Genetic Algorithm Optimization Toolbox*
GPU	*Graphics Processing Unit*
ICSP	*In-Circuit Serial Programming*
IDE	*Integrated Development Environment*
Kd	Ganho derivativo
Ki	Ganho integral
Kp	Ganho proporcional
LED	*Light Emitting Diode*
MATLAB	*Matrix Laboratory*
MOGA	*Multi Objective Genetic Algorithm*
MSE	*Mean Squared Error*
NPGA	*Niched Pareto Genetic Algorithm*
NSGA	*Nondominated Sorting Genetic Algorithm*
PID	Proporcional Integral Derivativo
POM	Problema de Otimização Multiobjetivo

PSO	*Particle Swarm Optimization*
PUC-Rio	Pontifícia Universidade Católica do Rio de Janeiro
PWM	*Pulse Width Modulation*
RMSE	*Root Mean Squared Error*
SF	Sistema Fuzzy
SFBR	Sistema Fuzzy baseado em regras
SFG	Sistemas Fuzzy Genéticos
SIF	Sistema de Inferência Fuzzy
SPEA	*Strength Pareto Evolutionary Algorithm*
UERJ	Universidade do Estado do Rio de Janeiro
USB	*Universal Serial Bus*
VEGA	*Vector Evaluating Genetic Algorithm*
ZDT	Zitzler Deb Thiele

SUMÁRIO

INTRODUÇÃO

A inteligência computacional é um conjunto de metodologias computacionais e abordagens que busca, através de técnicas inspiradas na natureza, o desenvolvimento de sistemas inteligentes que imitem aspectos humanos, tais como aprendizado, percepção, raciocínio, evolução e adaptação.

Devido aos bons resultados obtidos com a utilização das diversas técnicas envolvidas no campo de inteligência computacional, o número de pesquisas relacionadas vem crescendo ainda mais nos últimos anos. Além disso, a área de sistemas inteligentes é bastante ampla e abrange diversas aplicações (SANTOS, 2016) (FIGUEIREDO et al., 2014) (SANTOS et al., 2012) (LIMA, 2012).

Uma das justificativas para pesquisas na área de sistemas inteligentes, especialmente na descoberta de conhecimento, na mineração de dados e no aprendizado de máquinas, é a grande complexidade de modelagem de alguns sistemas e o imenso volume de dados digitais existente na atualidade que estão muitas vezes acima da capacidade humana de análise (SANTOS, 2014). Desta forma, a indução destes modelos pode ser feita de forma automática por meio de diversas abordagens tais como: redes neurais artificiais, métodos bayesianos, modelos gráficos e árvores de decisão, ou ainda através de sistemas com abordagens mais simbólicas, que além da capacidade de expressar o conhecimento de uma forma mais compreensível, possibilitam a introdução do conhecimento do especialista, como os sistemas fuzzy.

Os sistemas fuzzy são baseados na lógica fuzzy e são amplamente utilizados, especialmente em modelos de apoio a decisão e em sistemas de controle. Além disso, existem diversas aplicações relacionadas na literatura, como, por exemplo, na área de saúde e estudo da locomoção humana, em processamento de sinais da fala, no reconhecimento de informações e emoções, na economia e em sistemas de roteamento (LUCA et al., 2015). Sua característica de exprimir o comportamento de inferência humano possibilita um alto nível de compreensão, sendo a interpretabilidade um ponto forte dos sistemas fuzzy.

Dentre os pontos geralmente abordados na área de sistemas inteligentes, um ponto importante é a otimização, que consiste na busca da melhor solução para um

dado problema. Neste ponto, os algoritmos evolutivos são uma técnica de inteligência computacional usualmente utilizada devido à sua grande capacidade de busca. A otimização em algoritmos evolutivos consiste em tentar várias soluções e utilizar a informação obtida neste processo de forma a encontrar soluções cada vez melhores (LACERDA et al., 1999).

Inicialmente, a grande concentração de esforços na área de otimização consistiu em entender, desenvolver e aplicar métodos para a otimização de uma única função objetivo. No entanto, grande parte dos problemas reais de otimização envolvem objetivos múltiplos e não se pode aplicar a ideia de otimizar cada objetivo de maneira isolada. Cada objetivo tem seu grau de importância e muitas vezes os objetivos são conflitantes entre si, ou seja, a melhora de um objetivo provoca um pior resultado em outro (OLIVEIRA, 2005).

Em situações cotidianas é comum encontrar contextos que apresentam diversos objetivos. Como um simples exemplo, pode ser citado o processo para decisão sobre a compra de um carro em que é analisado o tamanho do carro, o consumo de combustível e o preço do carro. Outro exemplo é a procura de emprego em que vários pontos são considerados para realizar uma escolha adequada, como o salário inicial, a localização e oportunidades associadas. Em um meio industrial, geralmente, busca-se a maximização da qualidade de um produto ao mesmo tempo que o custo deve ser minimizado.

Motivação

Atualmente, existem diversas técnicas e algoritmos computacionais desenvolvidos para aplicação em problemas de otimização multiobjetivo (POM) motivados pela vasta área de aplicação. Grande número de trabalhos e pesquisas mostram bons resultados obtidos ao longo dos anos neste campo (FONSECA et al., 1995) (ALTINOZ et al., 2015) (JIANG et al., 2016).

As metodologias mais utilizadas incluem o uso de algoritmos genéticos e são baseadas no conceito da otimalidade de Pareto. Tal abordagem envolve a aquisição de uma fronteira com diversas soluções consideradas “ótimas” em relação aos objetivos analisados. Esta metodologia é caracterizada por ter uma articulação a-posteriori, ou seja, o processo de busca é executado de forma autônoma e após a

obtenção das soluções, um especialista deverá realizar uma escolha para decidir qual solução é a melhor para ser empregada ao problema. O processo de escolha da solução considerada aceitável, com um grande número de possibilidades e variáveis envolvidas, não é uma tarefa trivial e requer experiência e conhecimento do especialista.

Desta forma, a articulação de preferências do projetista feita a-priori, ou seja, antes da execução do algoritmo, e a utilização de uma técnica capaz de traduzir as preferências de forma mais simples e compreensível tornaram-se motivações para pesquisa e aplicação neste trabalho.

A proposta desenvolvida neste trabalho visa modificar a forma de avaliação tradicional de um algoritmo genético para possibilitar a avaliação de múltiplos objetivos. Para isso, foi escolhido utilizar um sistema fuzzy que agrega os diversos objetivos. O uso de sistemas fuzzy viabiliza avaliar simultaneamente todos os objetivos, integrando as preferências do usuário em relação a cada objetivo e a cada situação. Tal característica é uma vantagem em relação aos métodos multiobjetivos baseados na otimalidade de Pareto, pois não necessita da interferência do usuário para escolha da melhor solução ao final do processo, visto que as preferências são inseridas antes da evolução de forma mais simples e interpretável através da lógica fuzzy e assim o processo de evolução é guiado na direção das preferências e/ou especificações pré-estabelecidas.

A partir destes fatos este trabalho pretende contribuir com a apresentação de uma ferramenta de otimização baseada em algoritmos genéticos com avaliação multiobjetivo que utiliza um sistema de inferência fuzzy como agregador de objetivos. Tal ferramenta pode ser aplicada na resolução de diversos problemas multiobjetivos e apresenta a vantagem de não necessitar de intervenção do projetista ao final do processo de evolução para a escolha da solução mais adequada, pois o método de avaliação utilizando a lógica fuzzy permite inserir as especificações do problema de forma mais simples e interpretável quando comparado a outros métodos de otimização baseados em algoritmos genéticos.

Objetivos

A proposta deste trabalho é o desenvolvimento de um modelo evolucionário, baseado em algoritmos genéticos com a capacidade de avaliação de múltiplos objetivos, para otimização de sistemas e funções ou para viabilizar a criação de sistemas inteligentes, especialmente os sistemas fuzzy.

O foco principal abordado é a concepção de um modelo de avaliação capaz de considerar a otimização de mais de um objetivo no processo de avaliação de forma que as preferências e/ou especificações do projeto sejam inseridas no início do processo de uma forma mais simples e interpretável. Para isso, o método proposto para avaliação multiobjetivo utiliza a lógica fuzzy para agregar os objetivos.

O modelo implementado possui um ambiente de simulação que é utilizado para o projeto dos sistemas. Além deste ambiente, propõe-se a implementação de sistemas fuzzy em um ambiente real para aquisição de dados reais de entrada e saída, possibilitada por um protótipo de plataforma que realiza uma interligação entre o microcomputador e o hardware externo baseado em um microcontrolador.

Estrutura da dissertação

Esta dissertação está dividida em cinco capítulos.

No primeiro capítulo é realizada uma breve apresentação dos conceitos básicos de Sistemas Fuzzy e de Algoritmos Genéticos visando introduzir estes tópicos fundamentais do trabalho.

O capítulo 2 aborda a definição e principais características de problemas multiobjetivos, assim como algumas das diversas formas existentes na literatura para otimização de problemas deste tipo.

O capítulo 3 apresenta a plataforma evolutiva e o modelo híbrido proposto para avaliação de problemas multiobjetivos baseado em um sistema de inferência fuzzy, apresentando sua arquitetura e suas características.

No capítulo 4 são apresentados os estudos de casos desenvolvidos, as considerações e os resultados obtidos.

A conclusão deste trabalho é apresentada no capítulo 5 juntamente com sugestões de trabalhos futuros.

1 SISTEMAS FUZZY E ALGORITMOS GENÉTICOS

A busca pela compreensão e modelagem dos diversos sistemas que compõem a natureza é um tema que vem desempenhando um papel relevante na área da ciência e engenharia ao longo dos anos, especialmente no ramo de inteligência computacional.

Em sistemas complexos, o uso de modelos matemáticos se torna possível apenas através de grandes simplificações no próprio modelo, porém as simplificações implicam em resultados que, muitas vezes, não representam a realidade do sistema. Além disso, a capacidade de descrever o comportamento de alguns sistemas tende a reduzir, significativamente, de acordo com a complexidade do sistema (LEITE, 2009).

A inteligência computacional é uma área de pesquisa que trata de forma mais abrangente estes sistemas complexos e possui grande importância na atualidade em diversos campos de aplicação (EREMIA et al., 2016) (LABATI et al., 2016) (NEOCLEOUS et al., 2016). Ela engloba a computação evolutiva e os sistemas fuzzy, além de diversas outras técnicas, como as redes neurais artificiais, a inteligência coletiva, os sistemas imunológicos artificiais, entre outras.

Os sistemas fuzzy são atualmente utilizados na resolução de diversos tipos de problemas e têm demonstrado ao longo dos anos a sua eficiência em diferentes aplicações. Eles possuem a característica de representar situações reais em uma linguagem natural e interpretável. No entanto, a construção de sistemas fuzzy pode ser difícil em alguns casos em que não existe um conhecimento prévio e amplo do comportamento do sistema, dificultando a obtenção da base de regras e das funções de pertinência.

A partir desta necessidade, a união de técnicas de inteligência computacional, como redes neurais e algoritmos genéticos, com sistemas fuzzy tem se mostrado eficiente, pois integra a capacidade de aprendizado aos sistemas fuzzy. Estes modelos são chamados de sistemas híbridos.

Existem diferentes formas de sistemas híbridos fuzzy-genéticos. A utilização de sistemas fuzzy no processo de avaliação de algoritmos genéticos é um exemplo

de hibridização e é abordada mais detalhadamente na seção 2.5 e no capítulo 3 deste trabalho.

1.1 Sistemas Fuzzy

A lógica fuzzy ou lógica nebulosa é uma área de pesquisa intimamente relacionada à linguística e à ciência da cognição (ZADEH, 1965) (MAMDANI e ASSILIAN, 1975). Foi introduzida por Lotfi A. Zadeh na década de 1960 e é caracterizada por apresentar uma boa relação de compromisso entre significância e precisão, fornecendo um ferramental matemático para o tratamento de informações de caráter impreciso ou vago e auxiliar no controle e tomada de decisão, algo que os pesquisadores vêm buscando há um longo tempo. A lógica fuzzy permite lidar com informações imprecisas para tomada de decisão de problemas complexos e, assim, possui grande aplicabilidade em problemas de classificação, regressão, modelagem e controle inseridos em diversas áreas de conhecimento.

Os Sistemas Fuzzy (SF) são compostos por um conjunto de fundamentos que vão desde a teoria de conjuntos fuzzy, funções de pertinência, lógica fuzzy, passando pelas regras do tipo "*se-então*" e terminando nos Sistemas de Inferência Fuzzy (SIF).

Na teoria clássica de conjuntos, um elemento pertence ou não a um determinado conjunto. Na teoria de conjuntos fuzzy, essa caracterização é generalizada de forma que um elemento pode assumir infinitos valores no intervalo [0,1]. Um conjunto fuzzy *A* em um universo *U* é definido por uma função de pertinência (FP) e por um conjunto de pares ordenados, representando seus elementos. A função de pertinência é uma curva que associa cada elemento de um conjunto fuzzy a um grau de pertinência, μ_A, que indica o quanto determinado elemento pertence ao conjunto fuzzy. Dessa forma, um determinado elemento pode vir a pertencer a mais de um conjunto fuzzy com diferentes graus de pertinência (TANSCHEIT, 2007).

As funções de pertinência podem ter diversos formatos, sendo os mais comuns o formato triangular, trapezoidal, gaussiano, sino e sigmoidal. A escolha de

qual formato de FP deve ser adotado depende do contexto do problema estudado e da experiência e perspectiva do projetista. Levando em consideração apenas a teoria fuzzy, qualquer uma das funções de pertinência tem potencial para representar um conjunto fuzzy, porém, deve ser notado que cada uma das funções produz conjuntos fuzzy distintos e não existe uma regra única para o uso das funções de pertinência.

Um conceito importante na lógica fuzzy é o de variável linguística que é aquela cujos valores aparecem em sentenças na forma de linguagem "natural" representando as variáveis de um problema. Temperatura, velocidade, distância e altura são exemplos de variáveis linguísticas comuns em problemas de engenharia. Para cada variável linguística são associados termos linguísticos como: baixo, médio, alto, pequeno, grande e muito grande, que correspondem às características da variável linguística. A questão semântica é muito importante e uma associação feita de forma adequada possibilita um melhor entendimento do problema que está sendo analisado.

A Figura 1 apresenta um exemplo onde é avaliada a temperatura ambiente. Nele, funções de pertinência trapezoidais e triangulares representam os conjuntos correspondentes aos termos linguísticos *Muito Frio*, *Frio*, *Agradável*, *Quente* e *Muito Quente* da variável linguística *Temperatura* em um universo de discurso de 0 a 45°C.

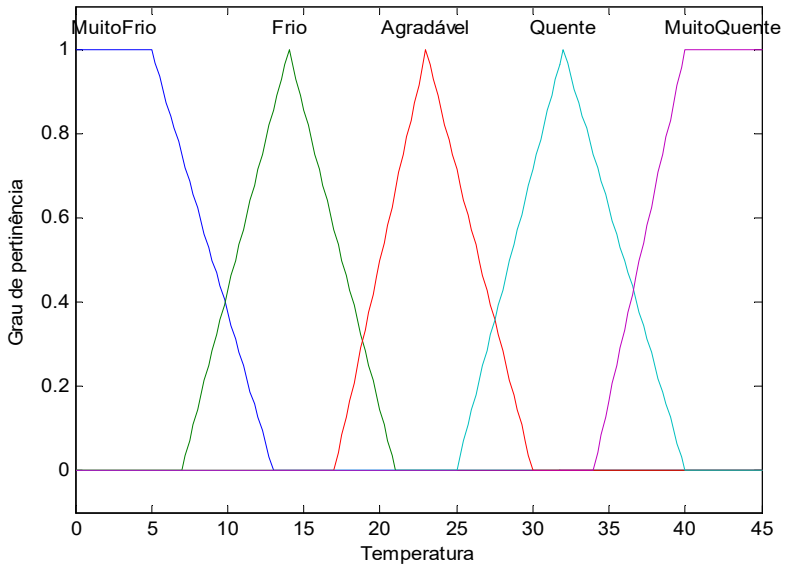


Figura 1 – Variável linguística “Temperatura”.

Com os conjuntos fuzzy é possível realizar operações que são utilizadas na inferência dos sistemas. Existem diversos operadores que podem ser empregados para as operações de união e interseção de conjuntos fuzzy, porém o operador mais empregado para a interseção é o mínimo, enquanto para a união é comum utilizar o operador máximo.

Uma ampla classe de modelos para operadores de interseção e união de conjuntos fuzzy é formada pelas normas triangulares ou t-normas e conormas triangulares ou t-conormas.

Uma t-norma é definida por uma função binária *t* $([0,1]x[0,1] \to [0,1])$ que satisfaz as seguintes condições $\forall x, y, z, w \in [0,1]$:

Monotonia	$t(x,w) \le t(y,z)$ para $x \le y$ e $w \le z$
Comutatividade	$t(x,y) = t(y,x)$
Associatividade	$t(t(x,y),z) = t(x,t(y,z))$
Condições limite	$t(x,0) = 0$ e $t(x,1) = x$

As principais operações t-norma empregadas em conjuntos fuzzy são o mínimo e o produto:

Mínimo	$\mu_{A \cap B}(x) = \min\left[\mu_A(x), \mu_B(x)\right]$
Produto	$\mu_{A \bullet B}(x) = \mu_A(x) \cdot \mu_{\mathbf{B}}(x)$

Uma t-conorma é definida por uma função binária *s* $([0,1]x[0,1] \to [0,1])$ que satisfaz as propriedades de monotonia, associatividade e comutatividade como a t-norma e nas condições limites apresenta o seguinte comportamento:

Condições limite	$s(x,0) = x$ e $s(x,1) = 1$

As operações t-conorma mais utilizadas são o máximo, a soma probabilística e a soma limitada:

Máximo	$\mu_{A \cup B}(x) = \max\left[\mu_A(x), \mu_B(x)\right]$

Soma probabilística $\mu_{A+B}(x) = \mu_A(x) + \mu_{\mathbf{B}}(x) - \mu_A(x) \cdot \mu_{\mathbf{B}}(x)$

Soma limitada $\mu_{A \cup B}(x) = \min(1, \mu_A(x) + \mu_{\mathbf{B}}(x))$

A Figura 2 ilustra o uso dos operadores de união (máximo) e de interseção (mínimo) aplicados a dois conjuntos fuzzy A e B.

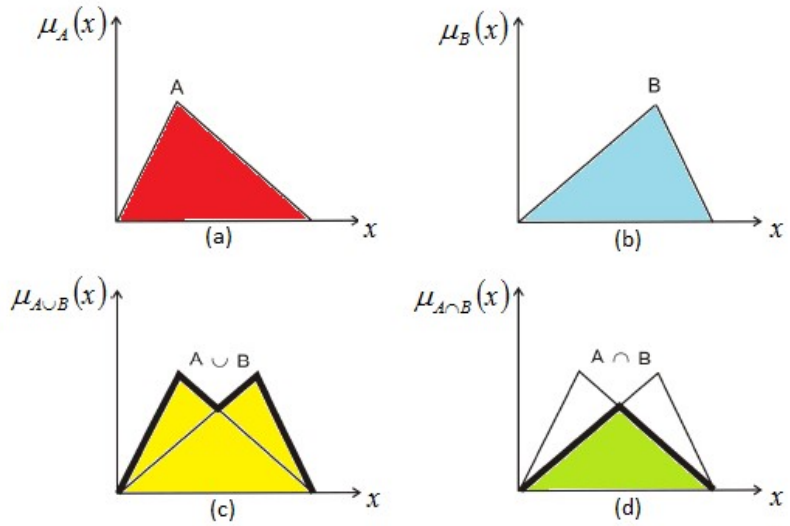


Figura 2 – Exemplo com os operadores máximo e mínimo.

Existem dois tipos de sistemas fuzzy: o Mamdani e o Takagi-Sugeno-Kang (TSK), que se diferenciam basicamente pelo tipo de consequente e processo de defuzzificação.

Os sistemas fuzzy baseados em regras (SFBR) utilizam uma estrutura de regras do tipo "*se-então*" capazes de descrever o comportamento de sistemas. Os antecedentes e consequentes das regras para sistemas do tipo Mamdani são proposições fuzzy que utilizam as variáveis e os termos linguísticos. Um exemplo de uma regra é mostrado a seguir:

SE *x* é baixo E *y* é alto ENTÃO *z* é médio

Onde *x* e *y* são variáveis de entrada, *z* é a variável de saída, "baixo", "alto" e "médio" são os termos linguísticos definidos por uma função de pertinência em *x*, *y* e *z*, respectivamente. O antecedente (premissa da regra) estima com que grau a regra

se aplica, enquanto o consequente (conclusão da regra) associa uma função de pertinência para uma dada variável de saída (AMARAL, 2003).

Um sistema de inferência fuzzy é um sistema que usa uma coleção de funções de pertinência e regras para inferir algo a partir de dados. Um SIF é formado pelo fuzzificador, um bloco de inferência, o defuzzificador e por regras conforme ilustrado no diagrama da Figura 3.

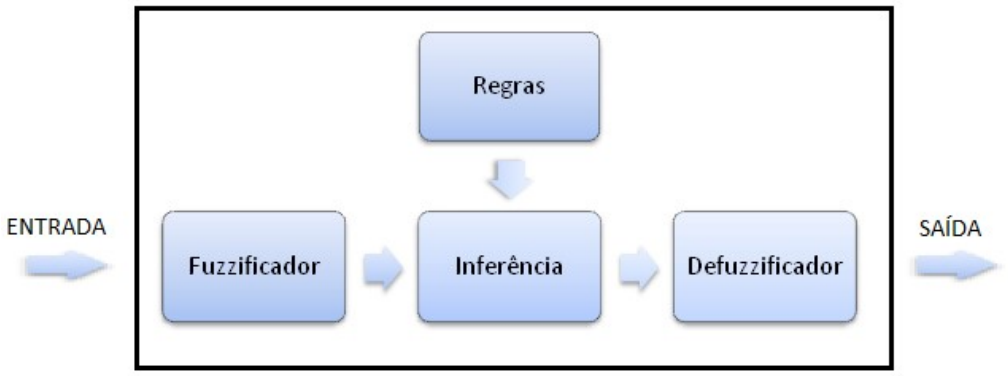


Figura 3 – Sistema de inferência fuzzy.

O bloco fuzzificador recebe os valores reais de cada entrada. Para cada uma delas, através das funções de pertinência definidas previamente, obtém-se o grau de pertinência com que a entrada pertence a cada conjunto fuzzy. As entradas do bloco são valores numéricos dentro do domínio da variável e a saída valores entre 0 e 1.

As regras fuzzy formam parte fundamental da estrutura de conhecimento em um sistema de inferência fuzzy. O bloco de regras fornece ao bloco de inferência as regras que descrevem o sistema. As regras podem ser obtidas através do conhecimento de um especialista ou podem ser extraídas de dados numéricos.

No bloco de inferência as regras são ativadas com as entradas fuzzificadas fornecidas pelo fuzzificador e é aplicado um processo de composição para gerar uma saída que será empregada como entrada no defuzzificador.

No modelo Mamdani, a saída *z* é obtida pela defuzzificação do conjunto fuzzy de saída resultante da aplicação de uma operação de t-conorma sobre os conjuntos dos consequentes, que, por sua vez, foram modificados via t-norma, pelo grau de disparo fornecido pelos antecedentes. Existem diversos métodos para

defuzzificação, os mais comuns são o centro de gravidade ou centroide, a média dos máximos e a média ponderada.

No modelo TSK o consequente de cada regra não é uma proposição fuzzy e sim uma função das variáveis de entrada. Geralmente, a função que faz o mapeamento de entrada e saída para cada regra, é uma combinação linear das entradas, isto é $z = px + qy + r$. No caso em que $p = q = 0$, temos $z=r$ (fuzzy singleton). A saída do sistema é obtida pela média ponderada (defuzzificação) das saídas de cada regra, usando-se o grau de disparo destas como pesos da ponderação.

1.1.1 Acurácia e interpretabilidade

Em sistemas fuzzy a acurácia e a interpretabilidade são dois pontos de elevada importância para avaliação da qualidade de um modelo (SHUKLA et al., 2012).

A acurácia é a característica que mostra a capacidade de representar o sistema real de uma forma fiel e pode ser definida pela relação de similaridade entre o sistema real e o sistema modelado. Existem algumas formas amplamente aceitas para avaliar quão boa é a acurácia. Uma possibilidade para esta avaliação é a porcentagem de padrões corretamente classificados, em um problema de classificação, ou o MSE (*Mean Squared Error*) para problemas de regressão.

A interpretabilidade descreve a capacidade que um sistema possui em representar o comportamento de um sistema real de forma compreensível. É uma propriedade subjetiva que depende da pessoa que faz a avaliação e está relacionada a vários fatores, como a estrutura do modelo, o número de variáveis de entrada, o número de regras fuzzy, o número de termos linguísticos e a forma dos conjuntos fuzzy. Não há uma medida padrão para avaliar a interpretabilidade de um sistema (GACTO et al., 2011).

A interpretabilidade e a acurácia são pontos contraditórios no desenvolvimento de sistemas fuzzy, pois uma melhoria em um aspecto, geralmente,

só acontece com a perda do outro. Esse compromisso entre acurácia e interpretabilidade é um motivador para diversas pesquisas nesta área.

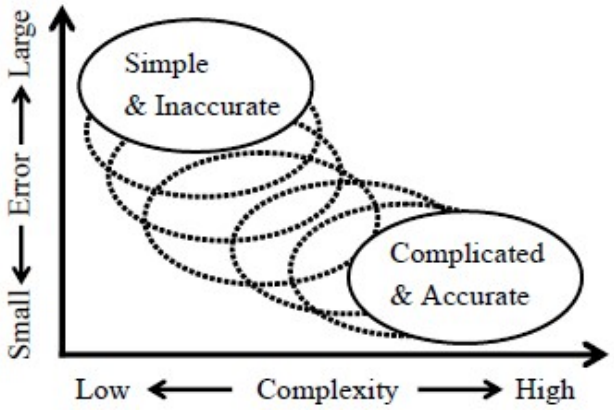


Figura 4 – Curva de compromisso entre acurácia e complexidade (ISHIBUCHI, 2007).

O compromisso entre estes critérios pode ser observado na Figura 4, que mostra uma curva com as melhores soluções, relacionando o erro e a complexidade entre elas. Observa-se que as soluções mais precisas, ou seja, com menor erro, são as mais complicadas, enquanto as soluções mais simples e de melhor compreensão possuem uma acurácia menor.

Desta forma, para certas aplicações é de extrema importância que um sistema fuzzy possua um bom desempenho aliado a uma boa interpretabilidade da sua base de conhecimento.

Em sistemas híbridos, que utilizam algoritmos evolutivos para geração da base de conhecimento fuzzy, podem ser aplicadas técnicas de evolução multiobjetivo possibilitando a obtenção de soluções que atendam aos dois critérios de acordo com as necessidades do usuário.

1.2 Algoritmos Genéticos

Os algoritmos genéticos (AG) compreendem uma área de pesquisa dentro da computação evolucionária. Eles são algoritmos matemáticos inspirados no princípio Darwiniano de evolução das espécies e da genética. Tais algoritmos usam o

conceito de que indivíduos mais aptos e adaptados ao ambiente possuem mais chances de sobreviver e deixar descendentes. Para isso, o algoritmo utiliza um mecanismo de busca adaptativa baseado no processo de evolução natural e recombinação genética.

A principal aplicação de algoritmos genéticos é em problemas de otimização com espaços de busca muito grandes ou complexos, o que inviabiliza o uso de técnicas tradicionais (AMARAL, 2003). Matematicamente, a otimização consiste em encontrar uma solução correspondente ao ponto mínimo ou máximo de uma função.

Os algoritmos genéticos privilegiam as melhores soluções encontradas a cada ciclo de forma a direcionar a busca para regiões onde é mais provável que estejam localizadas as soluções ótimas. Assim, o AG explora de forma inteligente as informações disponíveis para buscar melhores soluções para um determinado problema.

O funcionamento do algoritmo acontece através de um processo de evolução, em que cada iteração corresponde a uma geração. Uma população de indivíduos (possíveis soluções), identificados por cromossomos, são avaliados e associados a uma aptidão e submetidos a um processo de evolução, através de seleção e reprodução, durante várias gerações.

O fluxograma apresentado na Figura 5 mostra como ocorre a execução de um AG.

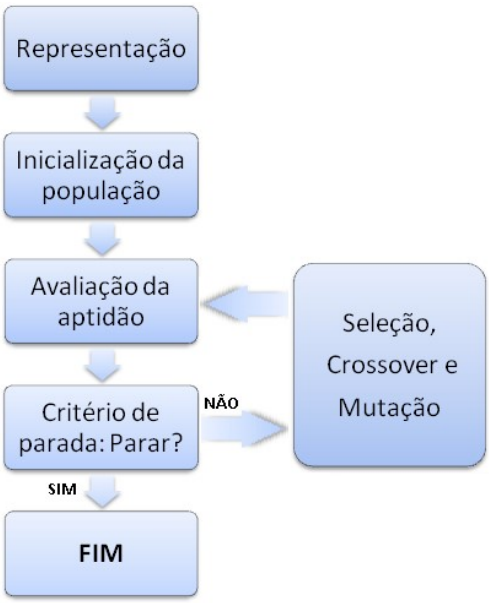


Figura 5 – Fluxograma de execução de um algoritmo genético.

Os aspectos que devem ser definidos para execução do AG são: a representação do problema e inicialização de uma população; a definição de uma função objetivo para avaliação dos indivíduos; e a escolha dos operadores genéticos e dos parâmetros gerais do AG.

A representação do problema corresponde ao mapeamento das possíveis soluções em uma estrutura de dados que possa ser manipulada computacionalmente. Tal estrutura é denominada cromossomo, o qual é composto por diversos genes. A representação pode ocorrer de duas formas: com tamanho fixo ou variável. Na representação de tamanho fixo, os cromossomos que representam os indivíduos da população possuem um número de genes constante. Por outro lado, na representação com tamanho variável, o número de genes pode variar ao longo do processo evolutivo, o que proporciona maior flexibilidade, porém necessita da inclusão de mecanismos de controle de tamanho adicionais. A representação é comumente realizada através de uma codificação binária (Figura 6), por ser de fácil manipulação e análise, porém também podem ser utilizados números inteiros ou reais (Figura 7).

0	1	1	0	1	0	0	1

Figura 6 - Cromossomo com representação binária.

3,2972	4,3241	7,1128	2,8972	1,0964	9,0089	5,4672	4,9421

Figura 7 – Cromossomo com representação real.

A inicialização da população corresponde à geração de uma população de *n* indivíduos que será a base para a evolução. A geração da população inicial é geralmente realizada de forma aleatória, porém também pode ocorrer de forma determinística dentro do espaço de busca.

A fim de avaliar a qualidade dos indivíduos, após cada nova população formada é aplicado a cada indivíduo um processo de avaliação e cada um recebe um valor de aptidão. A função objetivo ou função de avaliação de aptidão é definida com base na especificação do problema e é fundamental para uma execução bem-sucedida. Em geral, a função objetivo envolve apenas um único critério. Porém,

grande parte dos problemas reais envolve mais de um objetivo a ser considerado, desta forma a função objetivo deve utilizar métodos para conversão de uma medida de aptidão vetorial em um escalar (DAVIS, 1990).

Os três operadores genéticos empregados em AG são os operadores de seleção, *crossover* e mutação. São eles que proporcionam melhores soluções entre a população ao longo das gerações e possibilitam a evolução em problemas de otimização.

A seleção é o processo responsável por selecionar os indivíduos da população para formar uma nova geração de indivíduos. O operador exerce uma função similar a do processo de seleção natural biológica, em que indivíduos mais aptos possuem maiores chances de sobrevivência. Os métodos mais comuns de seleção são a roleta, o torneio e a normalização.

O método da roleta é o mais simples e o mais utilizado (GOLDBERG, 1989) e assemelha-se à roleta utilizada em jogos de azar. Cada indivíduo é representado na roleta conforme sua aptidão, isto é, indivíduos com aptidão alta ocupam uma parte maior da roleta que indivíduos com baixa aptidão, fazendo com que a probabilidade de seleção seja proporcional à aptidão de cada indivíduo. Após isso, gera-se um número aleatório no intervalo compreendido entre 0 e a soma das aptidões de todos os indivíduos. O indivíduo que possuir o número gerado é selecionado. O processo é repetido até a seleção de um determinado número de indivíduos requerido.

No método do torneio há uma escolha aleatória de um determinado número de indivíduos da população para participar de um torneio em que o indivíduo que possuir maior aptidão é selecionado para preencher a nova população. O processo é repetido *N* vezes até a seleção de *N* indivíduos.

Na seleção por normalização os indivíduos são ordenados de acordo com suas aptidões. Em seguida é realizado um escalonamento das aptidões, onde as aptidões originais são substituídas. Ao indivíduo de menor aptidão é atribuído o índice *1* e uma nova aptidão η^-, e ao indivíduo de maior aptidão o índice *N* e uma nova aptidão η^+. Assim, a probabilidade de seleção usando este mecanismo é uma função do índice do indivíduo e não da sua aptidão original como ocorre na seleção por roleta. De acordo com a função utilizada a seleção por normalização pode ser linear ou exponencial.

Além disso, o elitismo também é um importante recurso que está presente em grande parte dos algoritmos genéticos. Ele consiste em manter o(s) melhor(es) indivíduo(s) na formação da população seguinte. Esse processo contribui para o aumento do desempenho do algoritmo, pois evita que bons indivíduos sejam eliminados ou modificados pelos operadores de *crossover* e mutação.

O operador de *crossover* ou cruzamento possibilita a recombinação da informação de dois cromossomos diferentes a fim de gerar dois novos indivíduos. O resultado desta operação são indivíduos que, potencialmente, combinem as características dos indivíduos usados como base. O crossover pode ser realizado a partir de um ponto de corte ou mais.

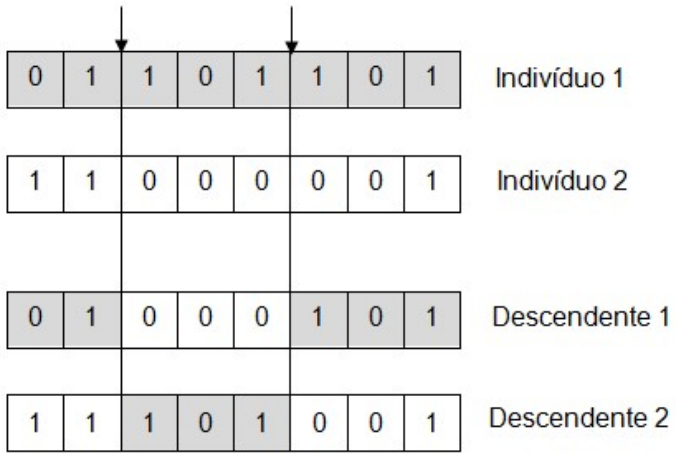


Figura 8 – Operador *crossover* em dois pontos.

A Figura 8 mostra o operador de cruzamento binário em dois pontos, em que um dos descendentes fica com as partes extremas do indivíduo 1 e a parte central do indivíduo 2 enquanto o outro descendente fica com as partes restantes.

O operador de mutação consiste na variação aleatória de um ou mais genes dos indivíduos e representa uma característica exploratória dentro do espaço de busca, pois insere novas características na população. A Figura 9 representa o operador de mutação aplicado ao terceiro gene de um determinado indivíduo.

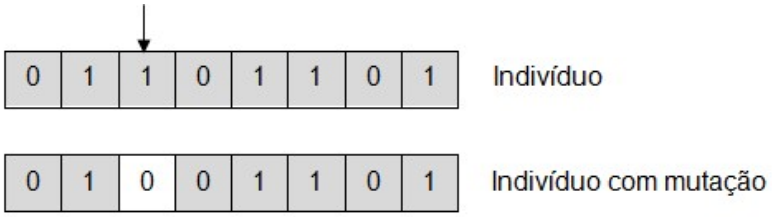


Figura 9 – Operador de mutação simples.

Os parâmetros gerais do AG exercem influência no seu desempenho e estabelecem um critério de parada para execução do algoritmo. Tais parâmetros incluem o tamanho da população, número máximo de gerações e taxa de aplicação dos operadores. A escolha dos parâmetros deve atender aos critérios empíricos estabelecidos ou às características específicas do problema em questão.

É possível destacar três principais vantagens dos algoritmos genéticos: a boa capacidade de busca global; a independência do domínio, pois trabalham sobre a codificação do problema possibilitando a elaboração de modelos para diferentes problemas; e a natureza paralela, que facilita a implementação em um processamento paralelo (ISHIBASHI, 2013).

Como principais desvantagens cita-se a possibilidade de obtenção de resultados sub-ótimos e o tempo de processamento que pode ser alto quando são utilizados muitos parâmetros.

1.3 Sistemas Fuzzy Genéticos

Os Sistemas Fuzzy Genéticos (SFG) constituem um dos métodos de hibridização mais conhecidos e sua implementação vem sendo estudada desde o início da década de 90 (HERRERA, 2008). A forma mais utilizada de união entre algoritmos genéticos e sistemas fuzzy é para a evolução da base de conhecimento fuzzy. Neste trabalho, esta forma de hibridização foi abordada como uma aplicação em um dos estudos de casos desenvolvidos. O algoritmo utilizado foi baseado no trabalho de (AMARAL, 2003), porém diversos outros trabalhos abordaram este tema através de diferentes algoritmos, entre eles pode-se citar (CINTRA et al., 2015) (ALCALÁ-FDEZ et al., 2011) (OSMAN et al., 2005).

O funcionamento de um SFG consiste basicamente de um sistema fuzzy acrescido da capacidade de aprendizado dos algoritmos genéticos. A Figura 10 mostra um diagrama geral de um SFG.

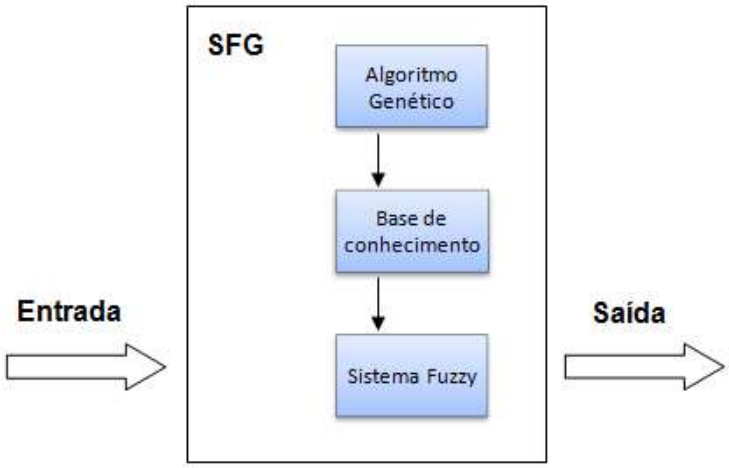


Figura 10 – Diagrama geral simplificado de um Sistema Fuzzy-Genético.

De uma maneira geral, o algoritmo genético pode ser responsável, em um caso mais simples, apenas por ajustar parâmetros do sistema fuzzy, ou, em casos mais complexos, por obter a base de conhecimento do sistema. As diferentes possibilidades de atuação do SFG estão ilustradas na Figura 11.

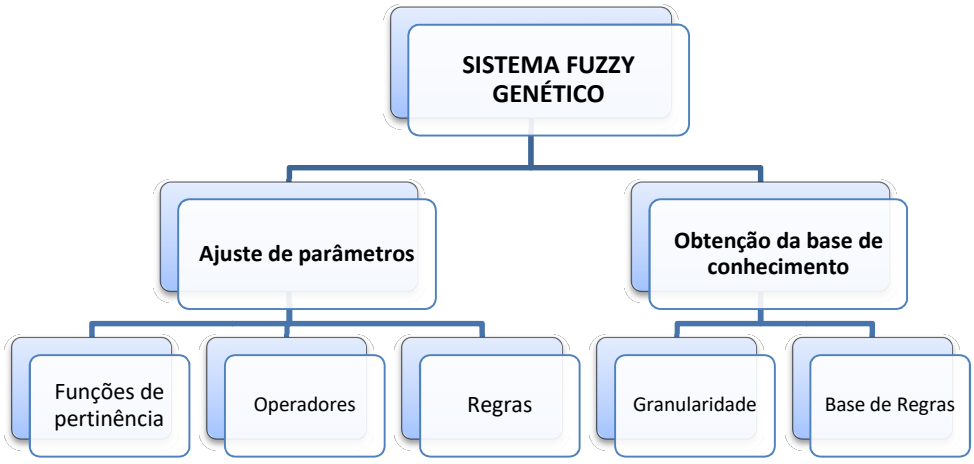


Figura 11 – Possibilidades de atuação de um Sistema Fuzzy-Genético.

Um ajuste de parâmetros pode ser empregado quando há um conhecimento a priori da base de conhecimento de um sistema fuzzy, obtido através de um

especialista ou por um método de extração de regras como o método de Wang e Mendel (WANG et al., 1992). O ajuste dos parâmetros é aplicado com o intuito de aperfeiçoar o desempenho do sistema, melhorando a acurácia ou reduzindo a base de regras. Basicamente, a otimização pode ocorrer através de três formas, conforme ilustrado na Figura 11: ajuste das funções de pertinência, ajuste dos operadores de fuzzificação e de defuzzificação e seleção de regras.

Nas três formas, os indivíduos da população do AG representam os parâmetros da base de dados que serão otimizados, para isso o cromossomo codifica os parâmetros associados às variáveis ajustadas. Por exemplo, no ajuste das funções de pertinência, cada indivíduo da população é capaz de representar todas as FP's do sistema. O número de genes do cromossomo é definido com base no tipo e no número de FP's associadas às variáveis linguísticas do SF. Normalmente, as FP's do tipo triangular são codificadas pelo valor dos seus três vértices e as gaussianas pelo valor médio e desvio padrão da função.

Outra aplicação do SFG é sua utilização para aprendizado da base de conhecimento de um sistema fuzzy, ou seja, o aprendizado da granularidade e o aprendizado da base de regras (BR) podem ser obtidos através de um processo evolucionário de busca.

O aprendizado da granularidade permite que o sistema identifique o melhor número e formato das funções de pertinência, enquanto o aprendizado da base de regras consiste na obtenção de todas as regras necessárias ao sistema fuzzy.

O uso dos AGs para a aprendizagem da base de regras pode ocorrer com base em diferentes metodologias, quatro delas são: o Método de Michigan, o Método de Pittsburgh, o Método Interativo e o Método Cooperativo e Competitivo (KOSHIYAMA, 2014). Os métodos se diferenciam pela forma de codificação da solução e pela forma de avaliação da qualidade da base de regras.

2 OTIMIZAÇÃO MULTIOBJETIVO

Comumente, em diversas situações cotidianas, são encontrados problemas que requerem a otimização de não apenas um, mas de vários objetivos simultaneamente. Estes problemas que requerem a maximização ou minimização de mais de um objetivo são chamados de problemas de otimização multiobjetivo (POM).

Podem ser citados diversos exemplos de problemas com múltiplos objetivos, como a compra de um carro em que a aquisição ótima é aquela que possui o custo mínimo e o melhor desempenho, ou em uma linha de produção industrial em que é desejado maximizar a qualidade do produto ao mesmo tempo da minimização do custo.

Em problemas multiobjetivos, geralmente, objetivos conflitantes devem ser analisados e uma única solução ótima raramente é encontrada. Existe um conjunto de soluções superiores que atendem em graus diferentes a cada objetivo. Assim, uma boa solução é aquela que não é dominada por nenhuma outra solução. Estas configurações superiores são chamadas de soluções não dominadas.

A otimização multiobjetivo é um importante campo de pesquisa não somente pela natureza da sua aplicabilidade na maior parte dos problemas reais, mas também pelas questões ainda em aberto. Entre elas pode ser citada a paralelização. Hoje em dia, o uso de computação em grade e Unidades de Processamento Gráfico (GPU – *Graphics Processing Unit*) abre novas e promissoras oportunidades para pesquisas nesta área, particularmente em relação à solução de problemas reais que possuem funções objetivo computacionalmente caras (LOPEZ, 2014) (LUONG et al., 2011) (ZHU et al., 2011).

A hibridização também é um campo a ser explorado. Nos últimos anos, algoritmos híbridos utilizando diferentes técnicas inteligentes têm sido propostos (MARTÍNEZ et al., 2013) (LARA et al., 2010). Estas abordagens podem ser combinadas com métodos alternativos para obter maior eficiência. A utilização de tais abordagens híbridas em aplicações no mundo real é promissora à medida que mais resultados de pesquisas se tornem disponíveis (COELLO COELLO, 2013).

2.1 Conceitos

A otimização multiobjetivo também é chamada de otimização multicritério ou de múltiplos objetivos e pode ser definida como um problema de busca por um vetor de variáveis de decisão que satisfaz às restrições e otimiza uma função vetorial cujos elementos representam as funções objetivo. Estas funções formam uma descrição matemática dos critérios de desempenho que são geralmente conflitantes uns com os outros. Assim, o termo "otimizar" significa encontrar uma solução que forneça valores aceitáveis ao usuário para todas as funções objetivo (OSYCZKA, 1985).

Desta forma, um problema de otimização multiobjetivo é definido por um conjunto de *k* funções objetivo e *n* variáveis de decisão. Matematicamente, podemos definir as equações (1) e (2):

$$F(X) = (f_1(X), f_2(X), ..., f_k(X))^T \quad sendo \quad X \in S \tag{1}$$

$$X = (x_1, x_2, ..., x_n)^T \tag{2}$$

Onde X é o vetor de variáveis, $F(X)$ é o vetor objetivo formado por k funções objetivo e S é o espaço de decisão.

A Figura 12 ilustra o espaço de decisão e objetivo em um problema com três variáveis de decisão e duas funções objetivo.

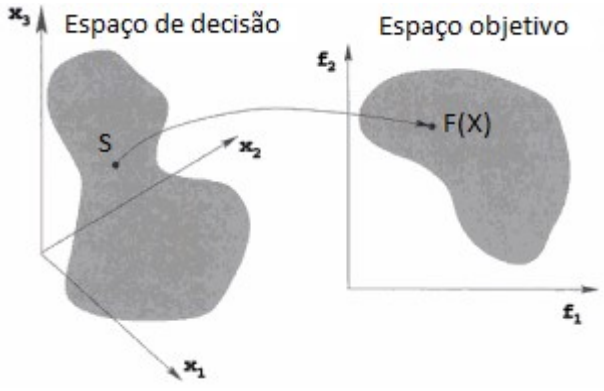


Figura 12 - Representação do espaço de decisão e espaço objetivo.

Diversas técnicas podem ser utilizadas para determinar quais pontos compõem a solução de um POM, porém um conceito importante e uma das abordagens mais comum é o conceito de dominância de Pareto.

2.2 Dominância de Pareto

A noção de soluções ótimas, generalizada por Vilfredo Pareto em 1896, se tornou conhecida como soluções Pareto ótimas (*Pareto-optimum*) e é bastante utilizada até a atualidade na resolução de diversos problemas de várias naturezas.

O método de Pareto emprega o conceito de dominância para comparar duas possíveis soluções do problema.

Dados dois vetores de variáveis de decisão quaisquer, $x, y \in S$, existem três possibilidades de classificação para os seus correspondentes vetores objetivo em um problema de minimização:

- Se $f_i(x) \leq f_i(y) \quad \forall i$ o vetor objetivo $f(x)$ domina $f(y)$ e o vetor de decisão x domina y;
- Se $f_i(x) \geq f_i(y) \quad \forall i$ o vetor objetivo $f(x)$ é dominado por $f(y)$ e o vetor de decisão x é dominado por y;
- Se $f_i(x) \leq f_i(y) \;\; e \;\; f_j(x) \geq f_j(y) \;\; i \neq j$ o vetor objetivo $f(x)$ é indiferente com $f(y)$ e o vetor de decisão x é indiferente com y e vice-versa.

  Onde $i, j \in [1,2,...k]$ e k= número de funções objetivo.

Assim, de uma forma geral, em um problema de minimização, diz-se que um vetor de decisão *x* domina um vetor *y* se, e somente se, as seguintes condições forem satisfeitas:

- O vetor objetivo $f(x)$ é menor ou pelo menos igual a $f(y)$ em todos os objetivos;
- O vetor $f(x)$ possui valor menor que $f(y)$ em pelo menos um dos objetivos.

Formalmente, define-se:

$$\forall i \in \{1,..., k\}, \quad f_i(x) \leq f_i(y) \ \cap \ \exists j \in \{1,..., k\}, \quad f_j(x) < f_j(y)$$

Para problemas de maximização as definições são análogas.

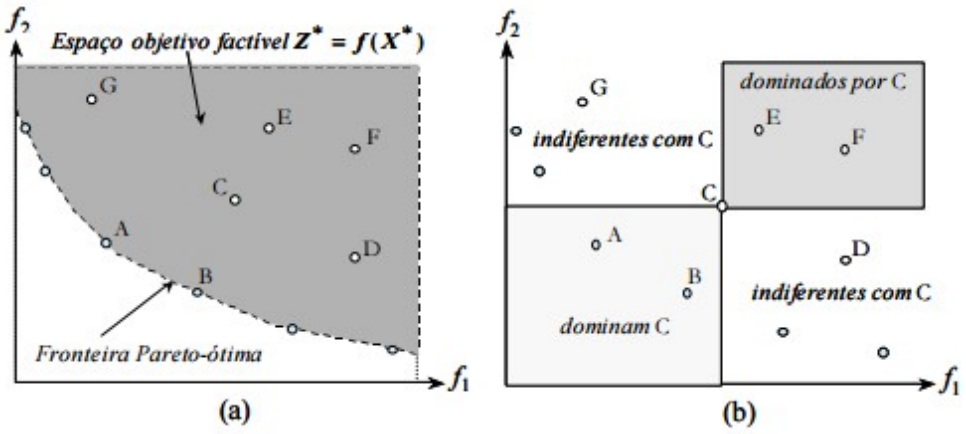


Figura 13 – Ilustração do conceito de dominância de Pareto (ARROYO, 2002).

Desta forma, diferentes pontos em um espaço de objetivos factível podem ser comparados e divididos em regiões de dominação. A Figura 13 (a) ilustra diferentes possíveis soluções em um espaço objetivo. Em (b) são mostradas as regiões de dominação em relação a um ponto *C*, para um POM de minimização. Por exemplo, os pontos *G* e *D* alcançaram uma melhor minimização de um dos objetivos, porém obtiveram valores mais elevados em outro, assim se tornam indiferentes. Os pontos *E* e *F* constituem soluções dominadas, enquanto os pontos *A* e *B* são soluções superiores e não dominadas.

A Figura 14 ilustra a comparação da distribuição de soluções dominadas, não dominadas e indiferentes em POM's com dois e três objetivos. A partir da figura é possível observar que a proporção da área de soluções indiferentes aumenta à medida que o número de objetivos analisados aumenta.

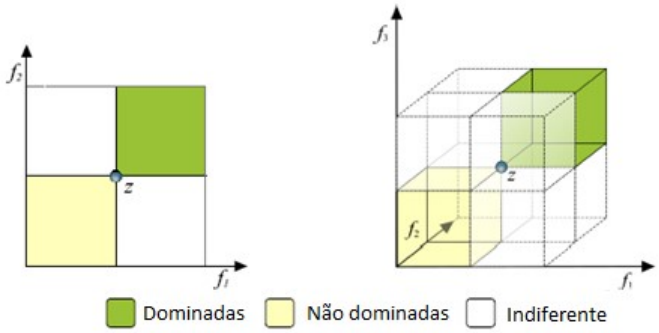


Figura 14 - Exemplo da localização e proporção entre soluções dominadas, não dominadas e indiferentes (adaptado de JAIMES et al., 2012).

A partir do conceito de soluções não dominadas é possível estendê-lo e caracterizar soluções Pareto-ótimas.

Assim, dado um conjunto de soluções factíveis, todas as soluções desse conjunto que são não dominadas formam o conjunto de soluções Pareto-ótimo.

Pode-se dizer que *X** é uma solução ótima de Pareto (Pareto-ótima) se não existe outra solução *X* que melhore algum objetivo sem simultaneamente piorar pelo menos um outro objetivo.

Todas as soluções Pareto-ótimas de um problema são indiferentes entre si e apresentam o melhor compromisso com as funções objetivo definidas. Na Figura 13 as soluções *A* e *B* não são dominadas por nenhum outro ponto do espaço, logo são soluções ótimas de Pareto.

A partir do conjunto de soluções Pareto-ótimo é obtida a fronteira ou frente de Pareto $\mathcal{P}$, definida por $F(X)$. Na Figura 13 (a) e na Figura 15 é possível observar a fronteira ótima de Pareto.

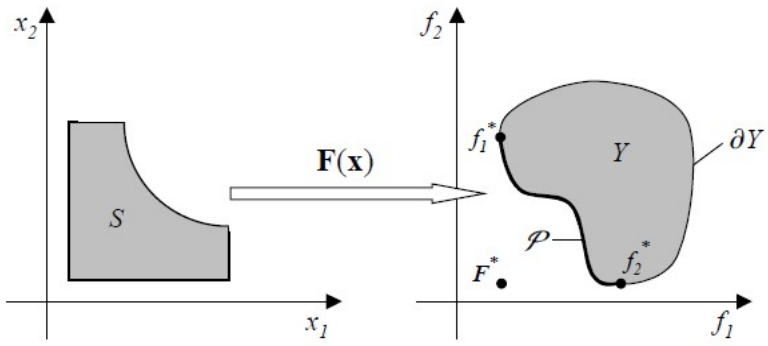


Figura 15 – Representação do espaço de soluções para um problema bi-dimensional (ANDERSSON, 2000).

Na Figura 15 também é possível observar os pontos ideais ou utópicos. Os pontos f_1^* e f_2^* pertencem à fronteira de Pareto e são as soluções ideais para a minimização individual dos objetivos f_1 e f_2, respectivamente. O ponto F^* é o ponto ideal que minimizaria os dois objetivos simultaneamente, ou seja, $F^* = \left(f_1^*, f_2^*\right)$. É importante observar que a existência desse ponto ideal seria a solução do problema. Porém, esta situação é extremamente improvável se o problema envolve objetivos conflitantes.

A Figura 16 apresenta as quatro possibilidades de fronteiras ótimas de Pareto para um hipotético POM com dois objetivos, f_1 e f_2, para cada caso de otimização possível das funções objetivo (minimização ou maximização).

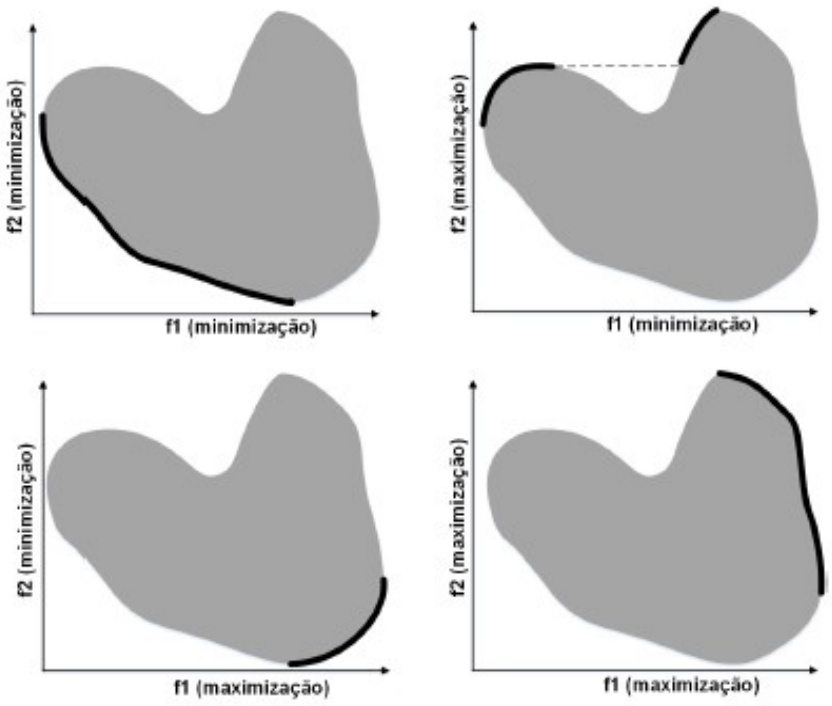


Figura 16 – Fronteiras ótimas de Pareto (DEB, 2001).

2.3 Técnicas tradicionais de otimização multiobjetivo

Existem diversos métodos utilizados para resolução de problemas de otimização multiobjetivo que podem ser classificados em quatro classes, diferenciadas pelo momento em que é realizada a escolha de preferência entre os objetivos. Essas diferentes classes e alguns exemplos de métodos de cada uma delas são mostrados na Figura 17 e descritos a seguir.

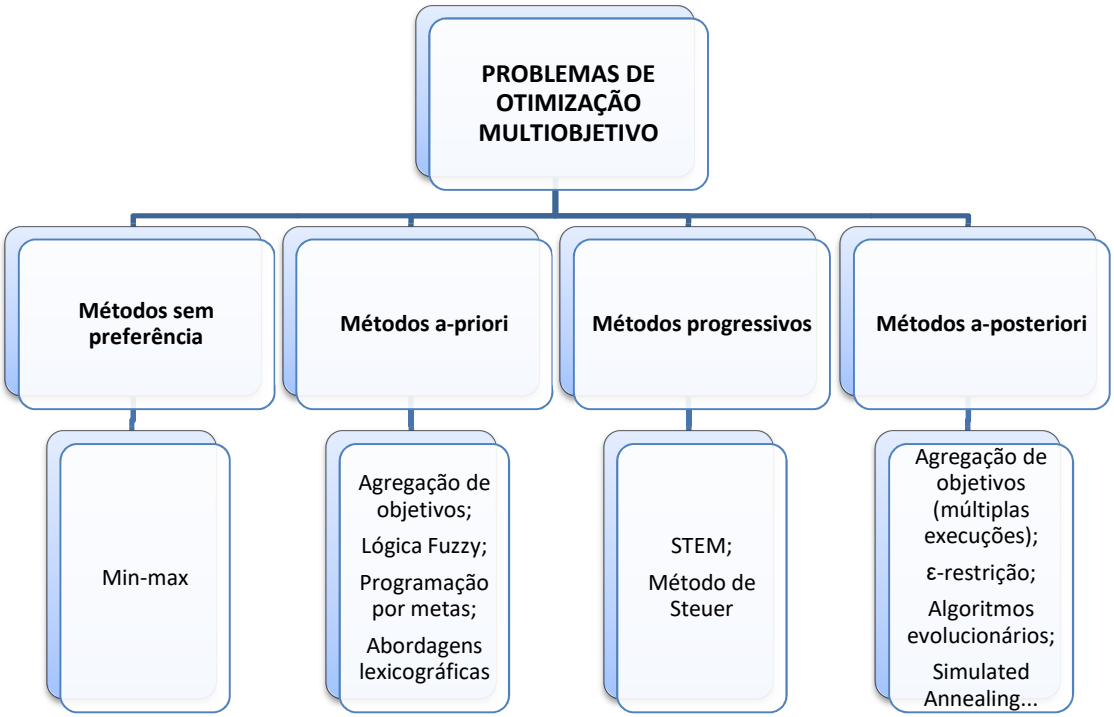


Figura 17 – Classificação de métodos de otimização multiobjetivo.

2.3.1 Métodos sem preferência

Esta divisão compreende métodos que não utilizam nenhuma informação a-priori ou a-posteriori sobre preferência entre os objetivos. A formulação min-max é um exemplo de método em que não é utilizada nenhuma informação sobre pesos ou preferência entre os objetivos e é brevemente apresentada a seguir.

2.3.1.1 Método min-max

O método min-max é baseado na minimização da distância entre uma possível solução e o ponto ideal *F** (Figura 15). A equação (3) descreve o método.

$$F = \left[\sum_{i=1}^{k} \left(\frac{f_i(x) - f_i^*}{f_i^*} \right)^p \right]^{\frac{1}{p}} \qquad (3)$$
$$x \in S$$
$$1 \leq p \leq \infty$$

Onde *S* é o espaço de soluções.

A variação do expoente *p* gera diferentes formas para o cálculo da distância. Os valores mais frequentes são 1 para formulações simples e 2 para distância euclidiana (ANDERSSON, 2000).

Através deste método, a solução encontrada representa um ponto na fronteira de Pareto que não utiliza qualquer informação prévia, sendo necessário que o usuário faça uma avaliação posterior sobre a qualidade da solução encontrada de acordo com o seu problema.

Este método pode ser utilizado em métodos interativos junto com outras técnicas, a fim de encontrar diversos pontos da fronteira de Pareto.

2.3.2 Métodos a-priori

Esta classe envolve métodos que utilizam informação prévia obtida do especialista, ou seja, de alguma forma é realizada agregação ou ponderação dos objetivos antes de realizar a otimização. Alguns desses métodos são descritos a seguir.

2.3.2.1 Agregação de objetivos

Um dos métodos mais simples, direto e mais utilizado é o método de agregação de objetivos. Neste método a avaliação de uma determinada solução é calculada pela média ponderada em relação a cada objetivo conforme a equação (4):

$$F = \sum_{i=1}^{k} \omega_i f_i$$
$$x \in S \tag{4}$$
$$\omega \in R^k \mid \omega_i > 0, \sum \omega_i = 1$$

Onde f_i representa a avaliação da solução em relação a um determinado objetivo i, e ω_i é o respectivo peso, para um total de k objetivos.

A simplicidade de implementação e possibilidade de inserção em técnicas tradicionais de otimização são algumas vantagens deste método. Por outro lado, um problema deste método é a dificuldade em determinar quais são os pesos adequados para cada problema. Em alguns casos é possível considerar os diferentes objetivos como sendo de mesma importância, e, portanto, os pesos iguais. Entretanto, na maioria dos casos a otimização de um determinado objetivo pode ser mais importante que a de outro e ser difícil a definição de quais são os pesos ideais. Logo, a aplicação do método de agregação de objetivos exige um custoso processo de ajuste, no qual diferentes conjuntos de pesos são testados até a obtenção de resultados satisfatórios (JONATHAN et al., 2010). Uma atualização dinâmica dos valores dos pesos seria uma forma de permitir um ajuste, ao longo do processo, de forma a atender todos os objetivos (ZEBULUM, 1999). Outra característica do método é a necessidade de normalização das avaliações a fim de obter valores comparáveis entre si.

2.3.2.2 Lógica Fuzzy

Em um problema de otimização multiobjetivo, a lógica fuzzy pode ser utilizada para associar a cada objetivo um valor de pertinência $\mu_i(f_i(x))$ que corresponde ao grau de satisfação do objetivo considerado, variando no intervalo entre 0 e 1. Para isso, as funções de pertinência devem ser definidas através do conhecimento de um especialista de acordo com o problema em análise.

Após obtenção da pertinência referente à cada objetivo, é necessário obter um valor único correspondente à uma avaliação geral de todos os objetivos. Deste modo, os valores de pertinência de cada objetivo são agregados através da aplicação de um operador de interseção e oferecem um valor único a ser otimizado. Assim, a função objetivo pode ser caracterizada de acordo com a equação (5) utilizando o operador mínimo ou (6) utilizando o operador produto.

$$F_{fuzzy}(x) = \min\left[\mu_1(f_1(x)), \mu_2(f_2(x)), \ldots, \mu_k(f_k(x))\right] \tag{5}$$

$$F_{fuzzy}(x) = \prod_{i=1}^{k} \mu_i(f_i(x)) \tag{6}$$

A lógica fuzzy ou os conjuntos fuzzy podem também ser utilizados com outros métodos. Por exemplo, juntamente a algoritmos genéticos, exercendo a função de avaliar e diferenciar possíveis soluções como apresentado em (CHAMANI et al., 2013), onde um algoritmo genético tem a função de buscar a melhor localização e o melhor tamanho de tanques de compensação. Neste caso, o sistema fuzzy usa a pressão e peso do fluido que será armazenado nos tanques como entradas e fornece um valor que é utilizado no processo de avaliação do AG.

De forma semelhante e baseando-se nos conceitos da lógica fuzzy e algoritmos genéticos foi desenvolvida a ideia deste trabalho, que é descrita em detalhes no capítulo 3.

Na seção 2.5 é abordada uma descrição mais detalhada de outros trabalhos existentes nesta área que utilizam a lógica fuzzy e algoritmos genéticos de uma forma diferenciada.

2.3.2.3 Programação por metas

No método de programação por metas (CHARNES et al., 1961) (PAL et al., 2008) (AZZABI et al., 2013) os objetivos são formulados por metas ou critérios a serem alcançados. Por exemplo, “o objetivo deve ser maior ou igual a *w*” ou “menor ou igual a *y*” ou “deve estar próximo a *z”*.

Alcançar um ponto ideal que satisfaça a todas as metas é geralmente impossível, assim o objetivo aqui é encontrar o ponto no espaço de soluções *S* que melhor se aproxime da solução ideal.

Existem diferentes métodos para cálculo deste ponto, um exemplo é a programação por metas ponderadas, que utiliza pesos para ponderar os desvios e distinguir os objetivos mais importantes. Outro exemplo é a programação por metas com priorização, em que as metas são agrupadas por prioridades e incorporadas ao modelo por meio de uma hierarquização nos objetivos, priorizando inicialmente a otimização do objetivo mais importante e seguindo para os demais nos estágios seguintes.

2.3.2.4 Abordagens lexicográficas

Em métodos com abordagem lexicográfica o especialista define uma ordem de importância para otimização dos objetivos (FERNANDES, 2015). Em uma situação com *k* objetivos, o objetivo mais importante é dito ser um objetivo de *prioridade 1* enquanto o menos importante tem prioridade *k*. Os objetivos mais importantes devem ser otimizados antes dos objetivos de menor importância. É um processo similar à uma ordenação alfabética, em que são comparadas as soluções em relação ao objetivo mais importante e em caso de empate o próximo objetivo é então avaliado. A grande desvantagem desse método é a possibilidade de ter objetivos não avaliados, portanto não é comumente utilizado.

2.3.3 Métodos progressivos

Os métodos conhecidos como progressivos ou interativos são utilizados em situações em que não há total determinação de pesos ou preferências entre os objetivos em um primeiro momento, mas ao longo do processo de otimização é possível que o projetista inclua informações. Nestes métodos, o responsável pela decisão intervém durante o processo de otimização articulando preferências e guiando a busca para regiões onde existem soluções de interesse.

2.3.3.1 STEM

O *Step Method* (STEM) (BENAYOUN et al., 1971) é um método interativo caracterizado pela redução do espaço de soluções a cada iteração. Em cada etapa é calculada uma solução que minimiza a distância em relação à solução ideal de forma semelhante ao método min-max.

Após cada iteração o decisor pode escolher a solução encontrada como solução final, caso os valores das funções objetivo sejam considerados satisfatórios, ou, caso contrário, pode diminuir o valor de alguma função objetivo a fim de tentar melhorar os outros objetivos que não possuem valor satisfatório. Desta forma, o algoritmo procede progressivamente reduzindo o espaço de soluções através da introdução de novas restrições em diferentes objetivos ao longo do processo de otimização.

2.3.3.2 Método de Steuer

Este método proposto por Steuer e Choo (1983), alcança um conjunto de soluções eficientes através de uma métrica de Tchebycheff ponderada

aleatoriamente, utilizando um subconjunto de soluções não dominadas, que está mais próximo de uma solução ideal.

Usando técnicas de filtragem, soluções altamente dispersas dos subconjuntos e subconjuntos menores de soluções não dominadas são apresentadas a cada iteração. O decisor interage com o algoritmo para guiá-lo no sentido de uma porção de um subconjunto não dominado que ele prefere.

O procedimento tem a vantagem de poder convergir para soluções finais não extremas. É especialmente adequado para a programação linear com múltiplos objetivos, mas é também aplicável em problemas não lineares.

2.3.4 Métodos a-posteriori

Esta classe de métodos compreende formas que permitem a obtenção de um conjunto de soluções ótimas de Pareto que são apresentadas ao decisor para uma posterior decisão sobre qual é mais adequada.

A grande vantagem destes métodos é o fato da solução não depender de preferências do decisor antes e durante a obtenção do conjunto de possíveis soluções. O conjunto ótimo de Pareto é obtido uma vez e diversas soluções podem ser consideradas a partir dele, de acordo com os objetivos necessários em cada situação a ser analisada.

Por outro lado, pode ser uma tarefa difícil para o decisor escolher qual a melhor solução em meio a diversas opções. Outra desvantagem é o grande custo computacional. A seguir são descritos alguns dos métodos mais utilizados.

2.3.4.1 Agregação de objetivos (múltiplas execuções)

A agregação de objetivos, quando executada diversas vezes para diferentes vetores de pesos, pode ser utilizada para encontrar diversos pontos na fronteira de Pareto. Trata-se de um método de busca exaustiva, sendo um método mais direto,

porém apresenta algumas desvantagens, pois dependendo das dimensões dos diferentes objetivos e da forma da fronteira de Pareto, é difícil selecionar a variação dos pesos para garantir que os pontos estejam espalhados uniformemente na fronteira. Outro problema ocorre quando o espaço de soluções não é convexo, impossibilitando a obtenção de todas as soluções ótimas de Pareto.

2.3.4.2 ε - Restrição

No método de **ε**-restrição (HAIMES et al., 1971) (RITZEL et al., 1994), um objetivo é selecionado para otimização e os demais são definidos como restrições. Por exemplo, em um POM com dois objetivos um valor ε_j é atribuído como restrição a um deles enquanto o outro objetivo é otimizado (7):

$$\begin{aligned} &\min \{f_i(x)\} \\ &\quad f_j \leq \varepsilon_j, j \neq i, j = 1...k \\ &\quad x \in S \end{aligned} \tag{7}$$

Alterando progressivamente os valores de restrição, diferentes pontos da fronteira de Pareto são encontrados. Através das restrições é possível obter pontos da fronteira de Pareto em espaços de soluções não convexos.

2.3.4.3 Algoritmos evolucionários multiobjetivo

Algoritmos evolucionários constituem um campo bastante eficiente na resolução de POM's. Uma variedade de técnicas utilizando algoritmos genéticos tem sido desenvolvida nas últimas décadas e uma breve descrição dos principais métodos encontra-se na seção 2.4. A grande vantagem obtida na utilização de AG's é o fato deles avaliarem simultaneamente um conjunto de possíveis soluções que permite encontrar o conjunto total de soluções da fronteira de Pareto em uma única

rodada do algoritmo sem que haja a necessidade da realização de diversas iterações como nos outros métodos (COELLO COELLO, 1999). Além disso, apresentam facilidade e flexibilidade de modelagem, são menos susceptíveis às características da fronteira de Pareto não convexa e descontínua e podem trabalhar em espaços de busca que são intratáveis pelas abordagens tradicionais.

2.3.4.4 Outros métodos

Diversos métodos tradicionais de busca utilizados em problemas de otimização simples podem ser adaptados e empregados em problemas com múltiplos objetivos.

Entre eles, pode-se citar:

- O *simulated annealing* ou recozimento simulado que é um método estocástico que se baseia na busca aleatória da vizinhança, inspirado no comportamento termodinâmico da matéria. Uma aplicação da técnica é abordada em (ACHARYA et al., 2015).
- A colônia de formigas, que se inspira no comportamento orientativo das formigas para encontrar o melhor caminho, onde cada indivíduo compartilha a informação acerca de seu caminho com os demais indivíduos. Prasad et al. (2013) abordam uma aplicação utilizando esta abordagem.
- Algoritmos meméticos, um método derivado dos algoritmos genéticos, onde a informação para busca não está somente no comportamento hereditário, mas também nas unidades de informação (memes), que são compartilhadas e enriquecidas por todos os indivíduos. Um trabalho relacionado pode ser encontrado em Rubio-Largo et al. (2016).
- Busca tabu, método de busca local que utiliza uma lista de movimentos proibidos, aceitando um indivíduo da vizinhança mesmo que ele degrade o valor da função objetivo. Como exemplo, pode-se citar o trabalho de García-Martínez et al. (2015).
- Procedimento aleatório adaptativo guloso, baseado na estratégia *multi-start*, ou seja, na repetição de melhorias sucessivas, que utiliza uma heurística

construtiva com aleatoriedade controlada para compor a solução inicial a cada iteração (MATEUS et al., 2010).

- Enxame de partículas (*Particle Swarm*), método semelhante ao de colônia de formigas, herdado dos modelos sócio-cognitivos aplicados ao aprendizado, tais como avaliação de estímulos, adaptação cultural e imitação. Uma aplicação para aquisição da base de conhecimento fuzzy usando enxame de partículas está em (MUKHERJEE et al., 2014).

2.4 Algoritmos Genéticos Multiobjetivo

Devido à importância e grande utilização de algoritmos genéticos na otimização multiobjetivo, as próximas seções abordarão mais detalhadamente alguns dos algoritmos mais utilizados.

2.4.1 VEGA

O primeiro algoritmo genético multiobjetivo foi o VEGA (*Vector Evaluating Genetic Algorithm*) desenvolvido por Schaffer (1985).

O algoritmo modifica apenas o mecanismo de seleção em relação ao AG original. Para isso, em um problema com *k* objetivos, a cada geração a população, de tamanho *M*, é dividida aleatoriamente em *k* subpopulações de tamanho *M/k* e cada uma é avaliada de acordo com um objetivo. Um operador de seleção proporcional é aplicado a cada subpopulação separadamente. Após a seleção, as subpopulações são unificadas e são executadas as etapas de cruzamento e mutação com a população completa, da mesma forma que em um AG de um objetivo.

A principal vantagem do algoritmo é a simplicidade e facilidade de implementação. Por outro lado, como cada solução é avaliada apenas em relação a um objetivo, as soluções tendem a aproximar-se do ótimo neste objetivo, assim a

população converge para um conjunto de soluções extremas com superioridade em um dos objetivos e avaliação ruim nos outros.

2.4.2 MOGA

No algoritmo MOGA (*Multi Objective Genetic Algorithm*) (FONSECA e FLEMING, 1998a e 1998b) é inserido o conceito de dominância e cada indivíduo é classificado de acordo com seu grau de dominância em relação aos demais.

A classificação de um indivíduo é dada pelo número de indivíduos que o dominam mais um. Os indivíduos na frente de Pareto não são dominados e, portanto, possuem classificação 1. Após isso, os indivíduos são ordenados e uma aptidão é calculada de acordo com uma função linear. Os operadores de seleção, cruzamento e mutação são empregados normalmente como em um AG tradicional.

A principal vantagem do MOGA é a simplicidade do método, que permite aplicação em outros problemas de otimização, como problemas combinatórios. Em contraste, embora o conceito de dominação seja usado, todas as soluções em uma mesma frente não dominada não possuem a mesma aptidão. Em particular, este algoritmo pode ser sensível à forma da fronteira de Pareto e à densidade das soluções no espaço de busca.

Trabalhos recentes nas áreas de redes de sensores sem fio (ELSERY et al., 2015), aprendizado de redes neurais (HAJIMANI et al., 2015) e veículos elétricos (SHAHVERDI et al., 2016) têm sido desenvolvidos utilizando o MOGA.

2.4.3 NPGA

Horn e Nafpliotis (1993) propuseram um algoritmo denominado *Niched Pareto Genetic Algorithm* (NPGA). Este é um algoritmo baseado na dominância de Pareto, porém não usa métodos de classificação, em vez disso, utiliza torneios de dominação para selecionar os indivíduos para a próxima geração.

Em cada torneio, um subconjunto aleatório da população, geralmente 10%, é usado como base para determinar o domínio dos dois competidores. Se um dos competidores é dominado por um membro do subconjunto, mas o outro não é, o não dominado é selecionado como vencedor. Se ambos ou nenhum são dominados, a seleção baseia-se na contagem de indivíduos semelhantes no espaço de atributo. Um indivíduo com poucos semelhantes é preferível a um indivíduo com um número elevado, a fim de manter a diversidade da população.

Como o algoritmo não aplica a seleção na população inteira, ele se torna rápido e produz soluções não dominadas. Por outro lado, sua principal desvantagem é o fato de requerer uma escolha adequada da quantidade de torneios para obtenção de um bom resultado.

Trabalhos existentes na área de energia (BENEDICT, 2014) e em redes de radares (SHI et al., 2014) baseiam-se no NPGA.

2.4.4 PAES

Pareto Archived Evolution Strategy (PAES) foi introduzido por Knowles e Corne (1999) e consiste em uma estratégia de evolução denominada (1 + 1), em que um único pai gera um único descendente, em combinação com soluções de um arquivo que registra as soluções não dominadas encontradas anteriormente. Este arquivo é usado como um conjunto de referência, para que cada novo indivíduo seja comparado e assim obtenha um caráter elitista.

Um aspecto interessante deste algoritmo é o procedimento utilizado para manter a diversidade, que consiste em um processo de aglomeração que divide o espaço objetivo em uma espécie de grade com hipercubos. O número de soluções situadas em cada hipercubo é calculado e caso o descendente esteja situado num hipercubo com menor número de soluções em relação ao hipercubo do pai, o descendente será escolhido para a próxima geração.

Uma vez que o processo é adaptável, é necessário um único parâmetro adicional para determinar o número de hipercubos. Além disso, o comprimento de cada hipercubo depende do conhecimento dos valores máximos e mínimos

possíveis para as funções objetivo. Uma desvantagem do algoritmo é a possibilidade de ele ficar trancado num ponto de máximo (ou mínimo) local, quando as mutações produzidas não forem suficientemente grandes para atravessar o espaço existente entre o ponto de máximo global e os diversos pontos de máximos locais.

2.4.5 NSGA

O NSGA (*Nondominated Sorting Genetic Algorithm*), proposto por Srinivas e Deb (1995), é uma modificação ao procedimento de classificação originalmente proposto por Goldberg (1989) e baseia-se no conceito de dominância.

Os indivíduos da população são classificados em diversos níveis. Antes da seleção a população é avaliada e todos os indivíduos não dominados são classificados em um mesmo nível com uma avaliação inicial alta. Após isso, para manter a diversidade, os indivíduos deste nível são diferenciados, recebendo valores de aptidão de acordo com o número de indivíduos dominados. Em seguida estes indivíduos são removidos temporariamente da população. O processo é repetido com a população restante até que toda ela seja classificada, sendo que os melhores indivíduos de cada nível obtêm uma nota inferior à menor avaliação do nível anterior.

Após a classificação de toda a população, os processos de seleção, crossover e mutação são aplicados.

Algumas desvantagens do NSGA, como a elevada complexidade computacional, a abordagem não elitista e a necessidade de especificação de um parâmetro para o cálculo da aptidão dentro de cada nível, impulsionaram a implementação de melhorias no algoritmo.

O algoritmo NSGA-II (DEB et al., 2002) possui características que minimizam essas desvantagens, entre elas a adição de um operador de seleção possibilitando a combinação entre gerações para selecionar os melhores indivíduos. O algoritmo basicamente constrói uma população de indivíduos concorrentes, classifica cada indivíduo de acordo com o nível de não dominação e aplica os operadores de evolução conforme o NSGA, porém, em seguida combina os pais e filhos antes de particionar a nova população em níveis. Outra característica do NSGA-II é o cálculo

da distância de aglomeração ou multidão (*Crowding Distance*) de cada membro, que representa uma estimativa da densidade de soluções ao redor de uma solução particular. Ele usa essa distância de aglomeração no seu operador de seleção a fim de manter uma população diversificada, certificando que cada membro possui uma boa distância de aglomeração. Isto mantém a população diversificada e ajuda o algoritmo a explorar todo o campo de soluções.

Em um trabalho mais recente (DEB et al., 2014) um algoritmo baseado no NSGA-II foi proposto para utilização em problemas com três a quinze objetivos, visto que os algoritmos anteriores trabalham melhor em problemas com poucos objetivos, geralmente dois ou três. O algoritmo denominado NSGA-III possui uma estrutura básica semelhante ao algoritmo NSGA-II, porém com mudanças significativas em seu mecanismo de seleção. Ao contrário do NSGA-II, que utiliza a distância de aglomeração, a manutenção da diversidade entre os membros da população no NSGA-III é auxiliada pelo fornecimento de uma série de pontos de referência bem espalhados. Estes pontos de referência podem ser predefinidos de forma estruturada ou fornecidos preferencialmente pelo usuário. Na ausência de qualquer informação de preferência, qualquer colocação estruturada de pontos de referência pode ser adotada.

O NSGA-II é amplamente utilizado na atualidade e podem ser encontradas pesquisas na área de roteamento de veículos (SHAMSHIRBAND et al., 2014), na área da engenharia elétrica (BUAYAI et al., 2016), (VERDEJO et al., 2015), otimização de sistemas inteligentes (OMAR et al., 2015) entre outras. O NSGA-III também já está sendo utilizado em novos trabalhos (LI et al., 2015).

2.4.6 SPEA

O SPEA (*Strength Pareto Evolutionary Algorithm*) é um algoritmo desenvolvido por Zitzler e Thiele (1999) e caracteriza-se por ser um algoritmo evolucionário multiobjetivo elitista com conceitos de não dominância. É definido, como o próprio nome sugere, pela força (*strength*) das soluções não dominadas.

O algoritmo funciona com a manutenção de duas populações, sendo uma população externa, que a cada geração armazena um conjunto de soluções não dominadas. Inicialmente, uma população combinada pela população corrente e a externa é construída. Em seguida, todas as soluções não dominadas nesta população recebem um valor de aptidão baseado no número de soluções que elas dominam.

Um destaque do método é a ausência de qualquer parâmetro pré-determinado e o fato da aptidão dos indivíduos ser determinada apenas pelas soluções armazenadas no conjunto de Pareto externo.

Zitzler et al. (2001) propuseram um desenvolvimento ao modelo original do SPEA e implementaram o SPEA2. As principais modificações realizadas no SPEA2 em relação ao SPEA são a utilização de uma função de aptidão baseada na quantidade de soluções dominadas e não dominadas de cada indivíduo; a incorporação de uma técnica de estimação da densidade de soluções vizinhas, através do cálculo das distâncias euclidianas entre cada indivíduo e todos os demais, o que permite uma orientação mais precisa do processo de busca; e um método de truncamento que permite a preservação de soluções nos limites do espaço de soluções.

Os bons resultados do método reforçam a importância do elitismo em problemas de otimização multiobjetivo. Atualmente diversos trabalhos utilizam o SPEA2 para otimização (MENG et al., 2016) (KHAJWANIYA et al., 2015) (SANTANDER-JIMÉNEZ et al., 2014).

2.5 Fuzzy Multiobjetivo

Alguns diferentes trabalhos utilizando a lógica Fuzzy têm sido propostos nos últimos anos e ela tem sido especialmente utilizada no processo de avaliação de indivíduos em algoritmos genéticos multiobjetivo.

Os conceitos da lógica fuzzy possibilitam originalmente a agregação de diversos objetivos em uma única avaliação. Alguns estudos demonstram que o uso da lógica fuzzy em problemas multiobjetivos tem apresentado bons resultados.

Em (HE et al., 2014) é desenvolvida uma nova métrica de desempenho para diferenciar indivíduos do conjunto ótimo de Pareto. Para isso foi adotada a lógica fuzzy e criada uma relação fuzzy de dominância de Pareto (FD – *Fuzzy Pareto Dominance*) que foi adicionada à estrutura do algoritmo NSGA-II e SPEA2.

Este método é especialmente aplicado em problemas com mais de cinco objetivos, pois em problemas deste tipo a maioria da população é composta por indivíduos indiferentes, ou seja, não dominados em relação aos outros. Assim, a aplicação de conjuntos fuzzy tem a finalidade de quantificar graus de dominação a esses indivíduos.

Para isso é realizada uma classificação entre os indivíduos da população, comparando dois indivíduos por vez. A função de pertinência aplicada em problemas de minimização é a metade direita de uma função Gaussiana, como ilustrado na Figura 18. O domínio varia entre -1 e 1, o que corresponde à diferença normalizada entre dois indivíduos em relação a um objetivo. Esta diferença é normalizada pelo valor absoluto da máxima diferença encontrada entre todos os pares de indivíduos em relação ao objetivo determinado.

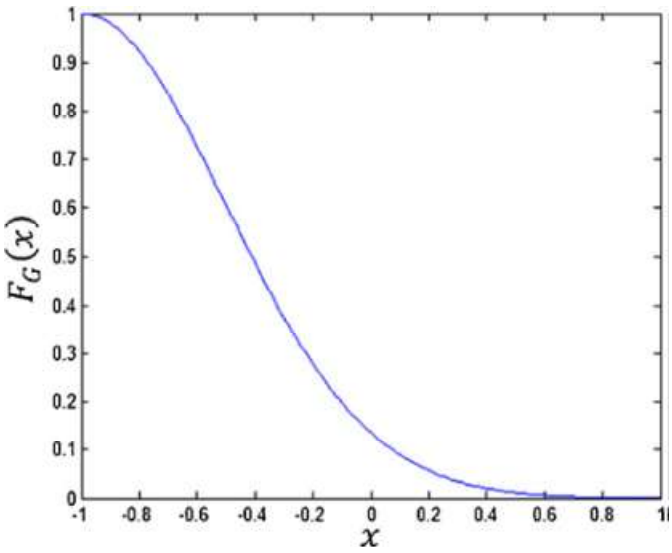


Figura 18 – Função de pertinência (HE et al., 2014).

Por exemplo, em uma comparação entre dois indivíduos, se em relação a um objetivo o valor de um indivíduo é muito menor que o do outro, a diferença normalizada estará próxima de -1 e a dominância deste indivíduo será alta comparada ao outro, com valor de pertinência próximo ou igual a 1. Por outro lado, caso esse indivíduo possua valor muito maior, a diferença normalizada resultará em um valor próximo a 1 e o grau de dominância será baixo, próximo de zero, indicando que esse indivíduo é dominado pelo outro neste objetivo.

Para o processo de agregação fuzzy, o desempenho de cada indivíduo em cada objetivo é considerado e os conjuntos fuzzy obtidos são combinados através da aplicação do produto como operador de interseção.

Assim, em um hipotético problema com dois objetivos, em que o conjunto fuzzy *A* representa o quanto o indivíduo *a* é melhor que o indivíduo *b* em relação ao primeiro objetivo e o conjunto fuzzy *B* representa o quanto o indivíduo *a* é melhor que o indivíduo *b* em relação ao segundo objetivo, a interseção entre o conjunto *A* e *B* corresponde a quanto o indivíduo *a* é melhor que o indivíduo *b* em relação aos dois objetivos do problema. Esse conceito pode ser estendido para problemas com *k* objetivos.

Este processo de comparação entre indivíduos possibilita atribuir diferentes graus de dominação aos indivíduos diferenciando uns dos outros. Para atribuição da aptidão, cada indivíduo é comparado aos outros indivíduos da população. De acordo com a dominação fuzzy Pareto é atribuído a ele um valor de dominação em relação a cada indivíduo, estes valores são depois somados e é atribuída uma aptidão única.

Comparando este critério fuzzy Pareto de dominância com a dominância original de Pareto, o fuzzy Pareto é mais eficaz, pois permite comparação em relação a todos os indivíduos de uma população ao mesmo tempo e através da diferenciação de avaliação permite um maior direcionamento durante o processo evolucionário.

Em (FARINA et al., 2004) foram apresentadas novas definições baseadas na lógica fuzzy para o conceito de otimalidade e dominação na tomada de decisão em problemas de otimização de muitos objetivos. Baseado nessas definições, diferentes subconjuntos das soluções Pareto ótimas podem ser formados utilizando informação provida pelo projetista para um parâmetro cujo valor varia entre 0 e 1. Quando o valor do parâmetro é zero, a definição de otimalidade é a mesma que o original de

Pareto. Quando este valor é alterado, subconjuntos de soluções ótimas podem ser obtidos correspondendo a diferentes graus de otimização.

Em (NASIR et al., 2011) é introduzido um conceito de dominância fuzzy no algoritmo MOEA/D (*Multiobjective Evolutionary Algorithm based on Decomposition*), que é uma técnica de otimização que utiliza a ideia de decomposição (ZHANG e LI, 2007).

O conceito de dominância fuzzy sugerido compara duas soluções entre os indivíduos da população atual e da nova população em relação a cada objetivo. Caso o novo indivíduo domine o anterior obtém grau de pertinência igual a 1, caso contrário obtém um valor de acordo com uma função exponencial, baseado na diferença entre eles. O produto é utilizado como operador de agregação fuzzy entre os conjuntos obtidos para todos os objetivos. O valor final μ obtido pela agregação fuzzy é comparado a um limiar de dominação τ. Caso μ seja igual a 1 ou maior que τ, a nova solução é considerada melhor que o indivíduo anterior e ele é substituído, caso μ seja menor, a solução é novamente comparada utilizando o conceito original de dominância do método de decomposição MOEA/D.

O conceito de dominância fuzzy foi aplicado neste método para corrigir a desvantagem que o método de decomposição original tinha em relação a soluções que estavam perto do vetor de pesos e eram negligenciadas, mesmo sendo capazes de gerar soluções melhores.

Köppen et al. (2005) propuseram uma extensão do algoritmo genético tradicional denominada *Fuzzy-Dominance-Driven Genetic Algorithm* (FDD-GA). O algoritmo utiliza a fuzzificação da relação de dominância de Pareto a fim de se obter uma representação numérica da relação de dominação entre dois indivíduos. São usados conceitos da lógica fuzzy para combinar os graus de dominância dentro de um conjunto de indivíduos. Valores são atribuídos a cada indivíduo e estes são classificados de acordo com o nível de dominação. Esta classificação é utilizada para avaliação e seleção dos indivíduos.

3 DESENVOLVIMENTO DO MODELO

A interpretabilidade característica de sistemas fuzzy é uma das grandes vantagens que impulsionam pesquisas neste campo. Tal fator possibilita inserir preferências e adaptar o sistema a diferentes situações utilizando uma linguagem natural e fácil de ser entendida.

A possibilidade de inclusão de entradas divergentes resultando em uma saída que atende a ambas, também é um ponto forte que permite sua utilização na resolução de problemas com múltiplos objetivos.

Desta forma, um método para avaliação através de sistemas fuzzy se tornou o foco na execução deste trabalho. Na literatura atual, outros trabalhos utilizam essa abordagem de uma forma diferenciada da proposta aqui, como foi abordado na seção 2.5.

A alta capacidade de busca dos algoritmos genéticos motivou a escolha desta técnica inteligente como base para utilização neste trabalho. Assim, foi desenvolvido um algoritmo genético capaz de obter uma solução de acordo com preferências estabelecidas conforme os diversos objetivos do problema, sendo que, para isso, utiliza-se um sistema de agregação fuzzy. O algoritmo e suas características são descritos neste capítulo.

Com isso, a grande vantagem do modelo desenvolvido é a inclusão de preferências e/ou especificações do usuário no início do processo de uma forma mais simples e interpretável, pois utiliza a lógica fuzzy para inserir essas preferências. Comparando o modelo à algoritmos que utilizam o conceito de Pareto, este fato é de grande importância pois evita que sejam apresentadas diversas soluções para escolha da melhor pelo projetista no final do processo.

Além disso, o modelo desenvolvido inclui um protótipo de plataforma com a finalidade de implementar sistemas inteligentes em um ambiente real. Para isso, é realizada uma interligação entre o microcomputador e um hardware que inclui um microcontrolador. O modelo e características desta plataforma também são descritos neste capítulo.

3.1 Características gerais do algoritmo

A metodologia utilizada no presente trabalho possibilita a evolução de sistemas para minimização ou maximização de funções, geração da base de conhecimento de sistemas fuzzy, ajuste de parâmetros de controle, entre outras aplicações que podem ser desenvolvidas.

O modelo desenvolvido utiliza um algoritmo evolutivo para busca da melhor solução para um problema com mais de um objetivo. O algoritmo evolutivo utilizado é um algoritmo genético baseado no GAOT (*Genetic Algorithm Optimization Toolbox*) (HOUCK et al., 1996) e executado no Matlab. Para simulações dos sistemas de controle foi utilizado o ambiente do *Simulink* integrado ao Matlab.

O algoritmo genético empregado no trabalho segue o modelo apresentado na Figura 19.

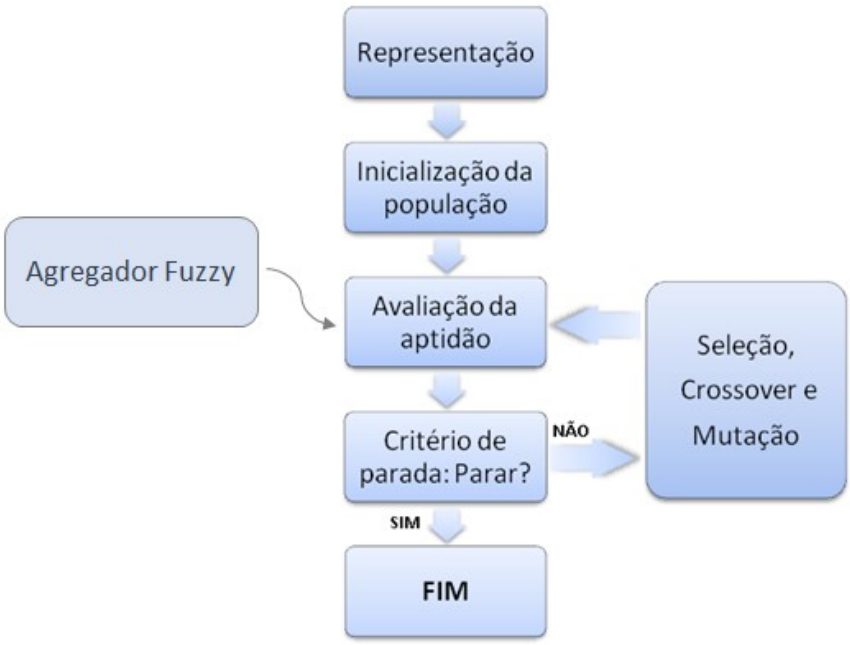


Figura 19 – Etapas de execução do algoritmo genético.

O algoritmo começa com uma população inicial que normalmente é gerada de forma aleatória, porém também pode ser gerada a partir de uma semente com soluções potencialmente boas obtidas a partir de outros métodos.

A avaliação tradicional de aptidão é realizada a partir de uma função *fitness* definida pelo projetista. Tal função gera um número escalar para cada indivíduo avaliado, que corresponde à aptidão do indivíduo em relação ao objetivo estabelecido pela função definida, logo, caso não seja aplicado nenhum método de agregação, possibilita avaliação de apenas um objetivo.

Em muitos casos, a função de aptidão corresponde ao cálculo do erro entre o resultado obtido para o indivíduo e a resposta esperada, que pode ser, por exemplo, o *MSE* (*Mean Squared Error*) ou o *RMSE* (*Root Mean Squared Error*). A análise de erro pode ser realizada através da comparação com uma base de dados pré-existente, que pode ser obtida através de cálculo, experimentalmente ou pela execução de um sistema real, ou ainda através de dados experimentais obtidos durante o processo de evolução. Por exemplo, em sistemas de controle onde a planta do sistema é conhecida, pode-se utilizar o *Simulink* para simular o comportamento do sistema e obter uma resposta que é utilizada no cálculo da aptidão do controlador proposto.

O processo de evolução inicia executando as operações de seleção, *crossover* e mutação. As taxas de execução destas operações são definidas pelo projetista antes do início do algoritmo. Da mesma forma, o tamanho da população e o número máximo de gerações são ajustados de acordo com a aplicação.

A evolução acontece até que um critério de parada seja alcançado. O critério de parada mais frequente é especificado por um determinado número de gerações. Outra possibilidade é estabelecer um valor a ser atingido ou parar a execução do algoritmo quando não há evolução por um determinado número de gerações.

O algoritmo genético utilizado apenas maximiza funções, portanto, em casos onde se deseja a minimização, usa-se a estratégia de inverter a função, ou seja, multiplica-se a função por *-1* e busca-se o ponto máximo desta nova função.

3.2 Agregador Fuzzy

A proposta desenvolvida neste trabalho visa modificar a forma de avaliação tradicional de um algoritmo genético para possibilitar a avaliação de múltiplos

objetivos. Para isso foi escolhido utilizar um sistema fuzzy capaz de agregar os diversos objetivos. O uso de sistemas fuzzy viabiliza avaliar simultaneamente todos os objetivos, integrando as preferências do usuário em relação a cada objetivo e a cada situação. Tal característica é uma boa vantagem em relação aos métodos multiobjetivos baseados na otimalidade de Pareto, pois não necessita da interferência do usuário para escolha da melhor solução ao final do processo, visto que as preferências e/ou especificações são inseridas antes da evolução de forma mais simples e interpretável através da lógica fuzzy e assim o processo de evolução é guiado na direção das preferências pré-estabelecidas.

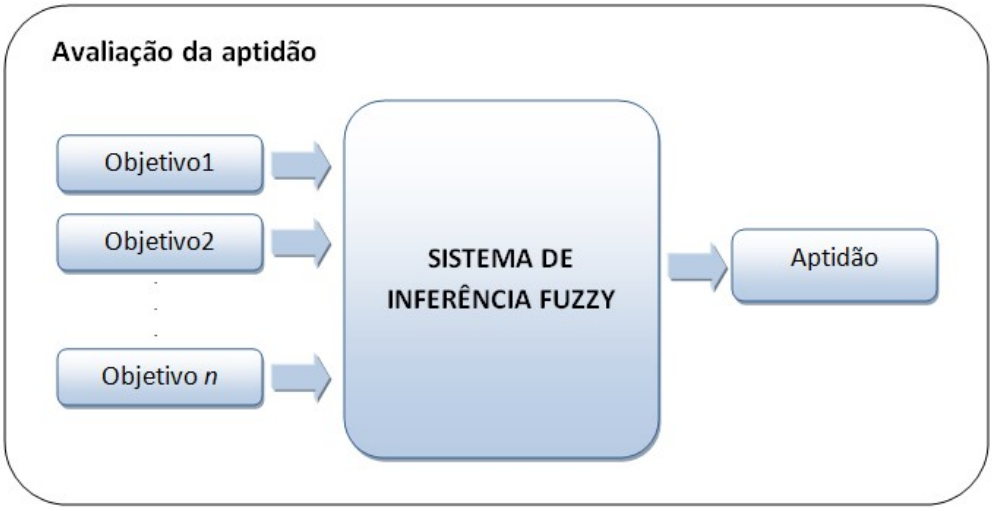


Figura 20 – Modelo de avaliação da aptidão com agregador fuzzy.

O sistema fuzzy responsável pela avaliação do algoritmo genético é denominado neste trabalho de agregador fuzzy. A Figura 20 ilustra o modelo da avaliação proposta.

Cada indivíduo da população do algoritmo genético representa uma possível solução do problema. Durante o processo de avaliação, os indivíduos são aplicados à função ou modelo que descreve o problema e os resultados obtidos em relação a cada objetivo são utilizados como entradas do sistema fuzzy. Para cada indivíduo da população é aplicado o agregador fuzzy resultando em um valor de aptidão.

O agregador fuzzy possui o funcionamento normal de um sistema de inferência fuzzy. Cada entrada do sistema corresponde a um objetivo e as funções de pertinência são formadas de acordo com o objetivo analisado.

As funções de pertinência possuem formato triangular e trapezoidal. Foi desenvolvido um modelo geral para agregação de dois objetivos, que pode ser usado como base para aplicação em qualquer problema. O modelo possui, para as entradas, cinco funções de pertinência triangulares distribuídas uniformemente dentro do intervalo de 0 a 1, correspondendo aos limites de variação de cada entrada que deve ser normalizada para facilitar e generalizar a aplicação, conforme Figura 21.

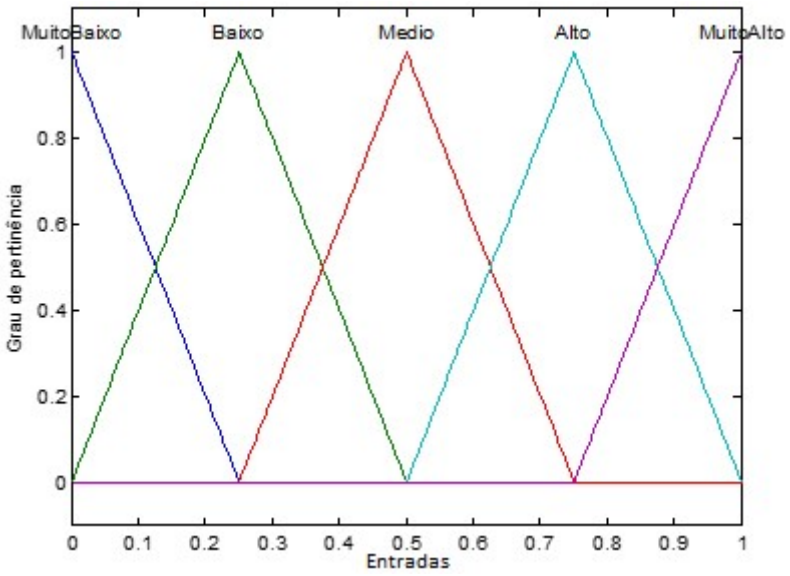


Figura 21 – Funções de pertinência base para entradas.

A partir deste modelo de funções podem ser feitas alterações de acordo com a aplicação. Por exemplo, ajustar o limite de alguma função de pertinência específica em uma determinada faixa desejada, reduzir o suporte ou alterar o valor em que a pertinência é máxima. Estes ajustes são realizados em casos em que há uma especificação bem definida ou é desejado melhorar o resultado obtido.

A saída defuzzificada do sistema fuzzy caracteriza a aptidão do indivíduo que está sendo avaliado. Para as funções de pertinência da saída é utilizado como padrão o formato mostrado na Figura 22, composto por cinco funções de pertinência.

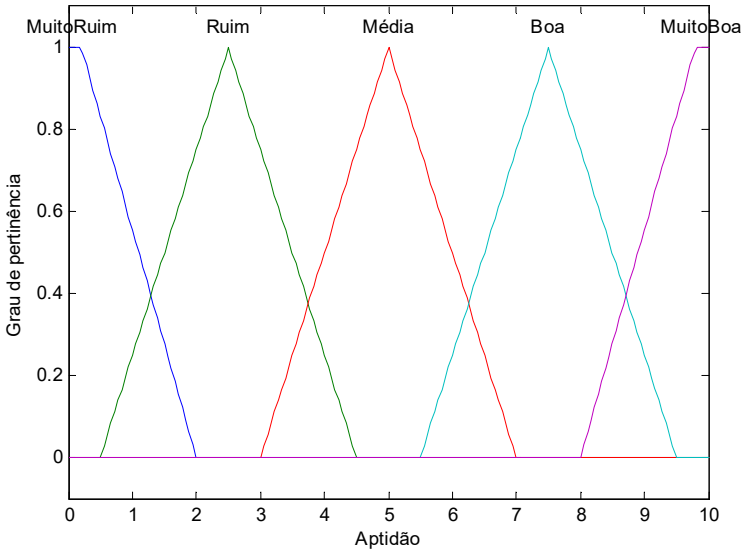


Figura 22 – Funções de pertinência base para saída.

O sistema fuzzy desenvolvido é do tipo Mamdani, caracterizado por ser mais simples e interpretável que sistemas do tipo TSK e todas as regras possuem mesmo grau de importância, ou seja, peso igual a um.

As regras do agregador fuzzy são elaboradas de forma a atender as preferências requeridas para o problema considerando cada objetivo.

Para exemplificar o processo de criação de regras, são mostradas na Tabela 1 regras básicas para minimização de dois objetivos sem preferência entre a minimização deles, ou seja, busca-se a minimização de ambos de forma igual.

Assim, quando as entradas correspondem a um valor Muito Baixo geram uma avaliação de aptidão Muito Boa. Da mesma forma, entradas com um valor Muito Alto possuem uma avaliação de aptidão Muito Ruim.

Tabela 1 – Modelo base para regras de minimização.

Entrada 1 / Entrada 2	*Muito Baixo*	*Baixo*	*Médio*	*Alto*	*Muito Alto*
Muito Baixo	Muito Boa	Muito Boa	Boa	Média	Ruim
Baixo	Muito Boa	Boa	Média	Média	Ruim
Médio	Boa	Média	Média	Ruim	Muito Ruim
Alto	Média	Média	Ruim	Muito Ruim	Muito Ruim
Muito Alto	Ruim	Ruim	Muito Ruim	Muito Ruim	Muito Ruim

Caso seja desejado priorizar a minimização de um objetivo em relação ao outro, as regras devem ser modificadas de forma a atender esta preferência. Da mesma forma, caso o problema envolva a maximização, as mesmas regras podem ser utilizadas, invertendo apenas os termos linguísticos dos antecedentes.

Os operadores utilizados no sistema são os operadores mínimo e máximo e a defuzzificação é realizada através do método do centro de gravidade.

Após execução da avaliação de todos os indivíduos da geração atual, o algoritmo genético continua o processo de evolução da forma tradicional, até a avaliação da geração seguinte, onde o processo de avaliação através do agregador fuzzy é novamente executado para todos os indivíduos, até que o critério de parada seja alcançado.

A técnica utilizada neste trabalho pode ser aplicada em diversos problemas e possíveis aplicações são abordadas no capítulo seguinte.

3.3 Interface Microcontrolada

A plataforma proposta prevê uma interface com um hardware microcontrolado e visa implementar sistemas inteligentes em aplicações reais. Um primeiro protótipo

simples foi concebido para realização de testes iniciais. A Figura 23 mostra um diagrama da plataforma proposta, ilustrando de uma forma geral as interligações existentes entre os componentes utilizados no protótipo:

- Um microcomputador (com Matlab e Simulink);
- Um microcontrolador Arduino UNO R3;
- Sensores e atuadores.

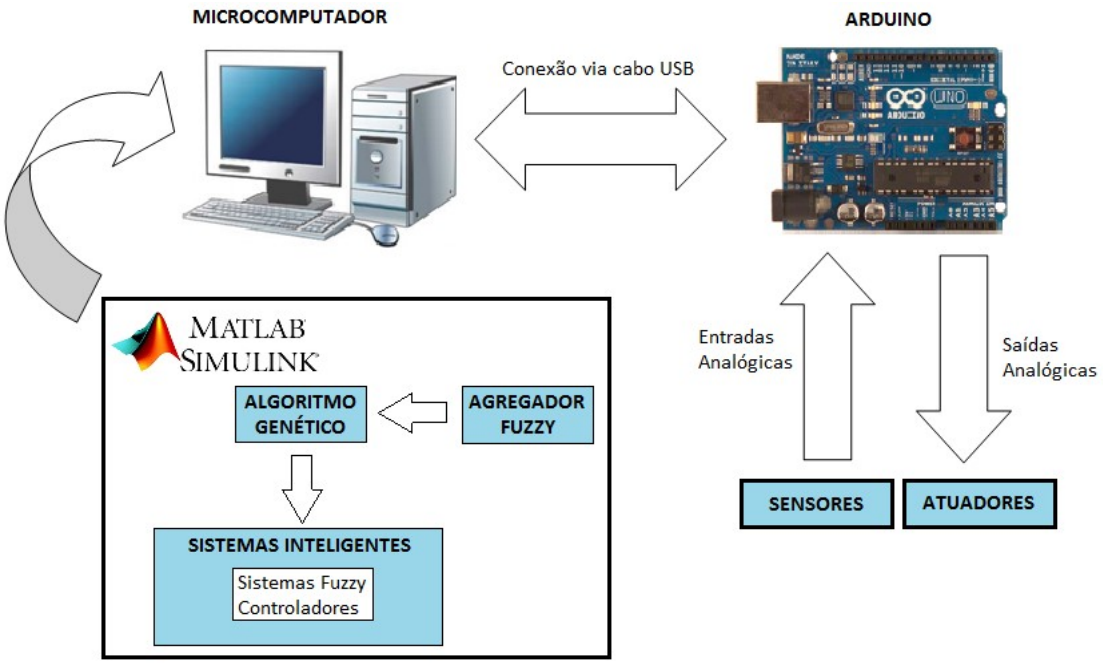


Figura 23 – Diagrama geral da plataforma.

O Arduino é uma plataforma de prototipagem acessível que viabiliza implementações eletrônicas e robóticas em projetos simples tipicamente usados para prova de conceito.

O hardware do Arduino UNO R3 consiste em uma placa de um microcontrolador baseado no ATmega328. Seu hardware possui 14 pinos digitais configuráveis de entrada ou saída e 6 destes pinos podem ser usados como saídas PWM (*Pulse-Width Modulation*) com uma resolução de 8 bits, 6 entradas analógicas de 10 bits, um oscilador de cristal de 16 MHz, um botão de reset e um LED. Para comunicação com o computador, possui conexão USB e um conector ICSP (*In-Circuit Serial Programming*) para gravação do microcontrolador de 6 pinos. Para a alimentação da placa, pode-se utilizar uma fonte de tensão externa na entrada de 7

a 12 V, podendo suportar um limite de tensão de 6 a 20 V, ou através do cabo USB (*Universal Serial Bus)* do próprio computador, porém nesse caso deve-se atentar para o limite de corrente das portas USB que é em torno de 500 mA. A tensão de operação dos componentes varia de 3,3 a 5 V e sua corrente de 50 mA a 40 mA.

O Matlab possui diversas bibliotecas e pacotes disponíveis em seu site oficial. Dentre eles, existe um pacote *open-source* denominado "*MATLAB Support Package for Arduino*" que foi utilizado neste trabalho e está disponível para *download* em duas versões. Para versões mais antigas do Matlab deve ser utilizado o pacote <*http://www.mathworks.com/matlabcentral/fileexchange/32374-legacy-matlab-and-simulink-support-for-arduino*> e para a versão R2014a ou versões mais recentes o pacote disponível em <*https://www.mathworks.com/hardware-support/arduino-matlab.html*>. Tal pacote possibilita a utilização do ambiente de programação do Matlab ou do Simulink para entrada e saída de sinais digitais e/ou analógicos de um sistema real através do Arduino via porta serial. O pacote consiste em uma interface de programação de aplicativos do Matlab instalada no computador e um programa servidor executado no Arduino. Desta forma, não é necessário utilizar a plataforma IDE do Arduino para sua programação, já que esse procedimento é realizado diretamente pela linha de comando do Matlab ou pelo Simulink (FONSECA, 2014).

Ao iniciar a utilização do Matlab juntamente com o Arduino é necessário especificar a porta a qual o microcontrolador estará interligado. Para aquisição e envio de dados existem funções próprias da biblioteca e comandos específicos que são utilizados diretamente na linha de comando. Além da linha de comando do Matlab, pode ser utilizado o Simulink para realizar a interface com o Arduino. Para isso, existe uma biblioteca incluída no pacote especificado anteriormente, constituída por diversos blocos que podem ser utilizados no Simulink junto com o Arduino. Tais blocos possuem a função de obter entradas analógicas ou digitais, enviar saídas analógicas ou digitais, configurar parâmetros de simulação, entre outras funções.

Por exemplo, para leitura de uma entrada analógica conectada ao Arduino utiliza-se o bloco mostrado na Figura 24 (a) ou o comando *a.analogRead*, especificando previamente os pinos do Arduino que serão utilizados. Cabe ressaltar que o comando ou o bloco *analogRead* retornam um valor entre 0 e 1023 gerado pela conversão analógico/digital, ou seja, o valor da entrada, que neste caso pode variar de 0 a 5 V, é convertido pelo conversor analógico/digital (A/D) de 10 bits

existente na placa do Arduino para uma escala correspondente de valores binários variando de 0 (0000000000) a 1023 (1111111111), logo, é capaz de capturar 1024 níveis discretos de um determinado sinal.

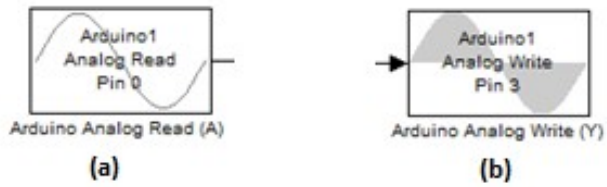


Figura 24 – Blocos para sinais analógicos do Arduino.

Da mesma forma, para realizar a escrita utiliza-se o bloco mostrado na Figura 24 (b) ou comando *a.analogWrite*. Em relação ao *analogWrite*, este envia ao Arduino um valor na escala de 0 a 255, obtido pela técnica de modulação por largura de pulso (PWM), de tal modo que *analogWrite*(255) solicita um ciclo de funcionamento a 100% (todo o tempo), *analogWrite*(127) um ciclo de trabalho de 50% (em metade do tempo) e *analogWrite*(191) um ciclo de trabalho de 75%, por exemplo.

Comandos e blocos semelhantes são utilizados para dados digitais (*a.digitalRead* e *a.digitalWrite*).

Neste trabalho, foi utilizado o Simulink para realizar a conexão com o Arduino por ser mais didático e possibilitar melhor implementação em sistemas de controle.

O microcomputador e o Arduino são conectados através de um cabo USB. Aos pinos de entrada e saída do Arduino são interligados os atuadores e/ou sensores necessários para o projeto a ser implementado.

Como mostrado na plataforma da Figura 23, um microcomputador é utilizado para executar o Matlab, evoluindo sistemas fuzzy ou controladores através do algoritmo genético com avaliação realizada pelo agregador fuzzy.

Após obtenção do sistema inteligente é necessário criar um projeto no Simulink que inclua este sistema e blocos capazes de obter e enviar os dados reais através do microcontrolador Arduino. Ao Arduino são interligados o(s) sensor(es) (potenciômetros, sensor de temperatura, sensor de luminosidade, etc) e o(s) atuador(es) (circuitos, módulo de potência, etc) para aquisição e envio de dados. A Figura 25 ilustra um exemplo de modelo criado no Simulink para controle fuzzy de um sistema real utilizando entradas e saídas do Arduino.

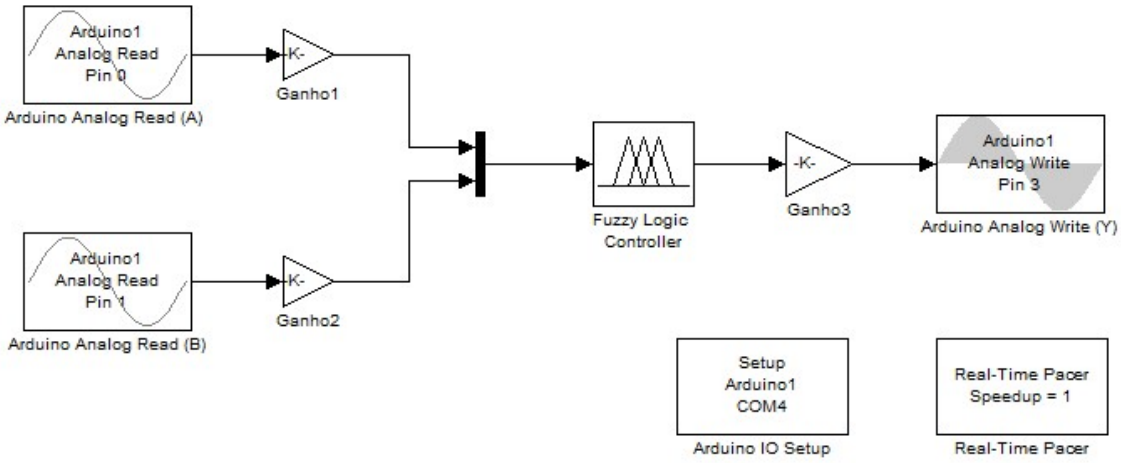


Figura 25 – Exemplo de modelo para controle no Simulink.

O modelo apresentado na Figura 25 inclui alguns blocos básicos: dois blocos para aquisição de entradas (*Analog Read*) com especificação dos pinos da placa Arduino utilizados; um bloco de controle que executa o sistema fuzzy desejado (*Fuzzy Logic Controller*); um bloco para a saída (*Analog Write*) especificando também o pino da placa ao qual a saída está interligada; blocos de ganho para compatibilizar os valores das entradas e saídas reais com a escala utilizada pelo controlador; e dois blocos de configuração da porta utilizada, velocidade de simulação e aquisição de dados.

Após conclusão do modelo no Simulink, com a porta e todos os pinos especificados, além da implementação do hardware e conexão do cabo USB entre o microcomputador e o Arduino, é possível iniciar a execução do modelo no Simulink e a aplicação real.

4 ESTUDOS DE CASOS

Foram realizados estudos de casos com diferentes aplicações que foram divididos em três conjuntos de aplicações separados de acordo com a finalidade, conforme descrito a seguir:

- Implementação de um modelo de sistema fuzzy com avaliação de dois objetivos.
    - Roteamento de redes de sensores sem fio: uma avaliação fuzzy multiobjetivo é aplicada na escolha do melhor caminho para transmissão de dados em redes de sensores sem fio.

- Implementação da técnica de agregação fuzzy na avaliação multiobjetivo de algoritmos genéticos.
    - Maximização de funções: é abordado um estudo que envolve a maximização de duas funções simultaneamente utilizando um algoritmo genético.
    - Minimização de funções: a técnica de agregação fuzzy é aplicada na minimização de problemas de benchmark multiobjetivo.
    - Síntese de sistemas fuzzy: o projeto de sistemas fuzzy por algoritmos genéticos é abordado através de dois estudos utilizando bancos de dados existentes na literatura.
    - Síntese inteligente de controladores PID: três diferentes controladores PID são evoluídos utilizando abordagens de avaliação mono objetivo tradicional e utilizando uma avaliação com agregação fuzzy considerando três objetivos.

- Implementação de um sistema fuzzy em um ambiente real.
    - Plataforma de implementação em hardware: é implementado um protótipo para aplicação em sistemas microcontrolados reais. Um estudo com obtenção de duas entradas reais e uma saída gerada através de um sistema fuzzy é apresentado.

4.1 Redes de sensores sem fio

Este primeiro estudo de caso é um típico problema multiobjetivo em redes de sensores sem fio. Este estudo não utiliza algoritmos genéticos, porém é apresentado neste trabalho com o objetivo de demonstrar de uma forma prática como funciona o método de avaliação fuzzy.

O objetivo do problema é avaliar e determinar o melhor caminho para transmissão de dados em uma rede de sensores sem fio, isto é, analisar a melhor rota para transmitir uma informação entre o sensor e o *gateway* considerando-se dois objetivos:

- Minimizar a distância percorrida pela informação entre o sensor e o *gateway*;
- Maximizar o tempo de vida útil das baterias dos sensores.

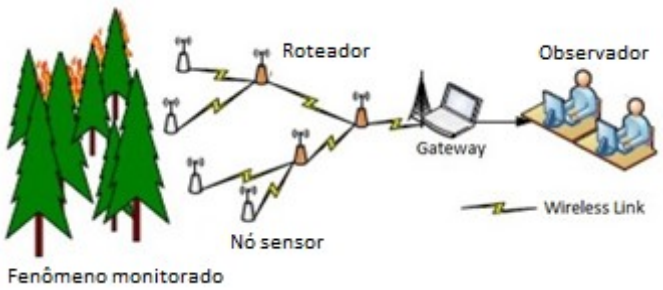


Figura 26 - Elementos de uma rede de sensores (adaptado de Coelho et al., 2017).

Redes de sensores sem fio são aplicadas em diversas áreas e necessitam ter um funcionamento eficiente com rápida transmissão de informações e prolongada vida útil. Para que a informação coletada pelos sensores seja transmitida ao *gateway*, é necessário, muitas vezes, que ela passe por outros sensores ou roteadores até chegar ao observador, que tomará uma ação dependente da informação coletada pelos sensores (Figura 26). Caso um roteador não esteja em perfeito funcionamento, a informação deve encontrar outro caminho para percorrer até chegar ao *gateway*.

Além disso, estes sensores e roteadores geralmente são operados por baterias que têm um período de vida útil limitado e torna-se impraticável trocar frequentemente esta fonte de energia. Para prolongar o período de utilização dos

sensores é interessante otimizar a transferência de dados de forma a poupar o uso desnecessário de roteadores e evitar falhas na transmissão de dados devido à falta de energia.

Um dos principais objetivos na transmissão de informações é garantir que os dados cheguem ao *gateway* no menor tempo possível, pois a cada roteador utilizado se gasta um tempo adicional. Uma forma de minimizar o tempo de atraso é utilizar um caminho mais curto, mais eficiente, no qual um número mínimo de roteadores seja utilizado para transmissão das informações.

Desta forma, um sistema de inferência fuzzy é projetado a fim de determinar para cada sensor ou roteador qual deve ser o próximo elemento a receber a informação que ele contém. Para isso, o sistema utiliza um agregador fuzzy que leva em consideração a distância que a informação percorrerá até chegar ao *gateway* e o estado de conservação do roteador, ou seja, se ele está em funcionamento e se possui uma boa carga de bateria.

Assim, um sistema de roteamento sem fio, no qual sensores realizam a medição de grandezas físicas e transmitem os dados coletados por eles até o *gateway* através de uma rede de roteadores e sensores, foi simulado no Matlab.

Inicialmente foi desenvolvido um algoritmo capaz de gerar uma configuração de rede simulando a localização dos sensores/roteadores em uma área industrial. Os nós sensores podem monitorar diversas grandezas físicas como temperatura, pressão e vazão e todos são considerados também roteadores, assim todos os pontos podem ser utilizados para o roteamento. Cem nós foram distribuídos em uma área quadrada com 1 km² e o *gateway* foi colocado no centro do ambiente a fim de evitar grandes distâncias entre ele e alguns roteadores. O alcance de transmissão de cada roteador foi definido como sendo de 200 metros.

A princípio são calculados quais os nós que podem ser alcançados por cada um dos sensores/roteadores da malha. Esta informação será utilizada para definir quais nós roteadores serão avaliados pelo sistema fuzzy. A escolha dos nós a serem utilizados na transmissão da informação coletada pelos sensores será realizada de acordo com as avaliações obtidas pelo sistema fuzzy e é feita com base no nível de bateria e na distância entre os nós que se encontram dentro do seu espaço de alcance e o *gateway*. O nó mais próximo do *gateway* e com maior nível de bateria recebe uma avaliação maior e é definido como prioritário. Assim, evita-se que haja

alguma falha na transmissão por conta da falta de energia, ao mesmo tempo em que, na transmissão da informação estaria sendo utilizado um número mínimo de componentes no processo.

Cada sensor ou roteador que possuir uma informação para ser transmitida executa um sistema fuzzy utilizando os parâmetros dos roteadores que podem receber a informação, ou seja, que estão dentro do alcance do sensor transmissor. O sistema fuzzy possui como entradas o nível de bateria dos roteadores que estão no alcance e a distância entre estes roteadores e o *gateway*.

A saída do sistema é uma avaliação da qualidade de cada nó para receber a informação. Após execução do sistema fuzzy, é realizada uma classificação entre todos os nós disponíveis e aquele com maior nota é o escolhido para dar continuidade na transmissão.

Na implementação do sistema fuzzy, foram utilizadas funções de pertinência com formato triangular e trapezoidal. As funções de pertinência foram definidas dentro dos limites de cada variável de entrada e saída. Por exemplo, a bateria pode variar entre 0 e 100%, assim, foi definido que um nível de bateria abaixo de 35% começa a ser considerado baixo e abaixo de 20% é totalmente baixo. A bateria acima de 65% começa a ser um nível alto e acima de 90% é totalmente alta. Valores em torno de 50 % são considerados médios conforme ilustrado na Figura 27.

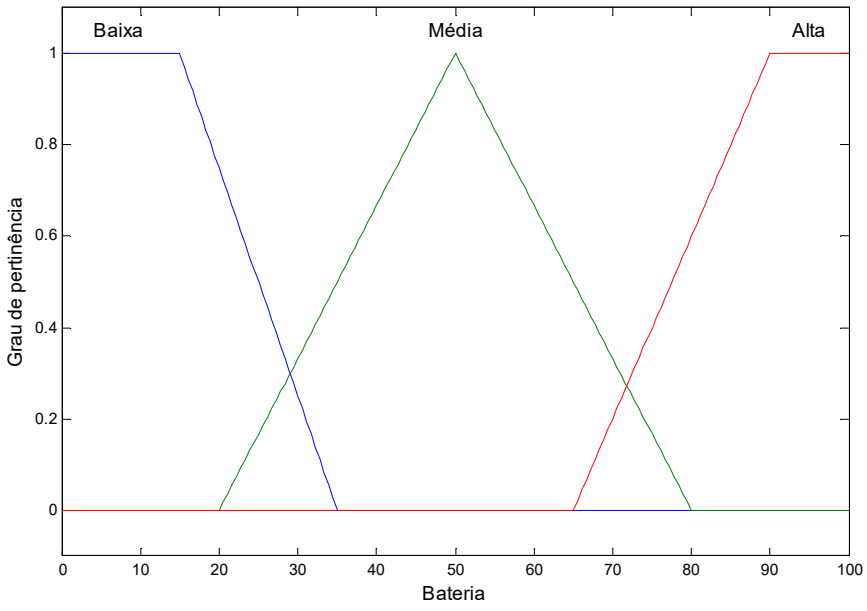


Figura 27 – Funções de pertinência da variável de entrada Bateria.

Da mesma forma, são criadas as funções de pertinência para a distância, ilustradas na Figura 28.

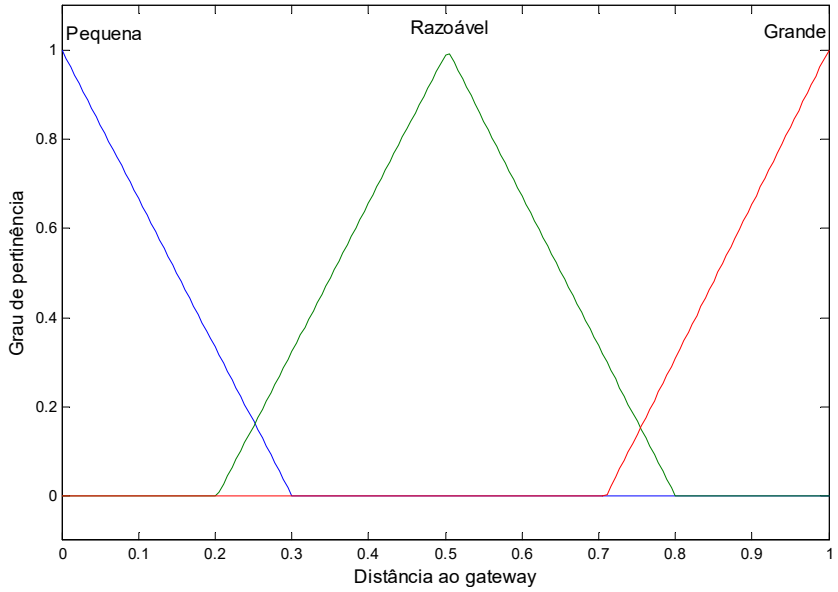


Figura 28 - Funções de pertinência da variável de entrada Distância.

A saída corresponde a uma avaliação do sensor de acordo com as entradas. Foram criadas três funções de pertinência dentro de um intervalo de 0 a 10, representando uma avaliação *ruim*, *média* e *boa*, conforme Figura 29.

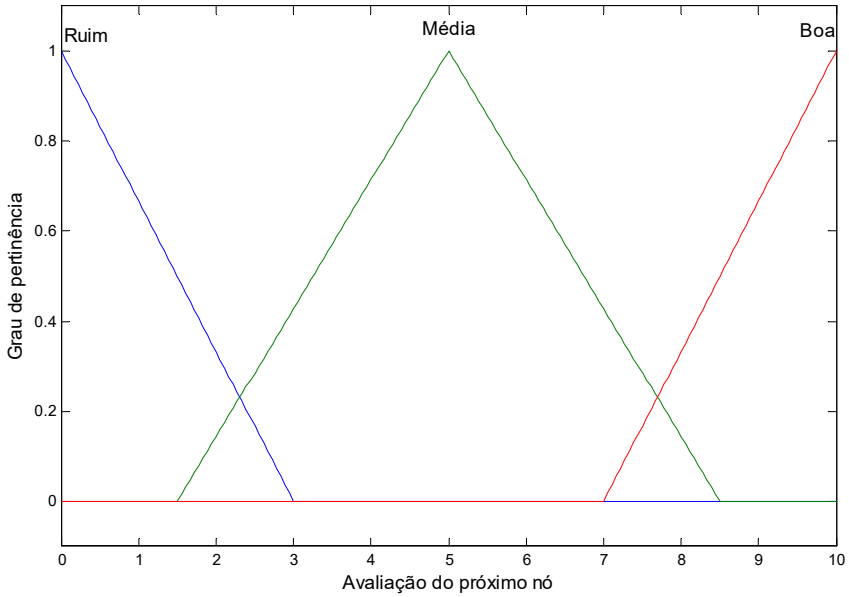


Figura 29 - Funções de pertinência da variável de saída Avaliação.

As regras foram criadas de forma a obter uma avaliação *Boa* quando a bateria é *Média* ou *Alta* e o nó possui uma distância *Pequena* em relação ao *gateway*. Uma distância *Grande* e bateria *Baixa* caracterizam uma avaliação *Ruim*. As demais regras foram criadas seguindo o mesmo raciocínio e estão indicadas na Tabela 2.

Tabela 2 – Regras para avaliação do roteamento.

Bateria / **Distância**	*Baixa*	*Média*	*Alta*
Pequena	Média	Boa	Boa
Razoável	Ruim	Média	Média
Grande	Ruim	Ruim	Média

A fim de verificar a eficácia do sistema de avaliação fuzzy implementado, foram realizadas algumas simulações.

Inicialmente, foi definido qual seria o nó sensor de onde partiria a informação. A partir disso executou-se o sistema fuzzy para escolher o nó para o qual a informação seria direcionada. O nó com melhor classificação foi escolhido e a informação transmitida a ele. Após isso o sistema fuzzy é executado novamente, utilizando agora as informações referentes a este novo roteador. Este processo é repetido sucessivas vezes até que o *gateway* seja alcançado.

Em uma primeira simulação foi escolhido um sensor no canto inferior direito. O caminho obtido para transmissão está indicado na Figura 30. A quantidade de bateria existente nos roteadores próximos também está indicada na figura.

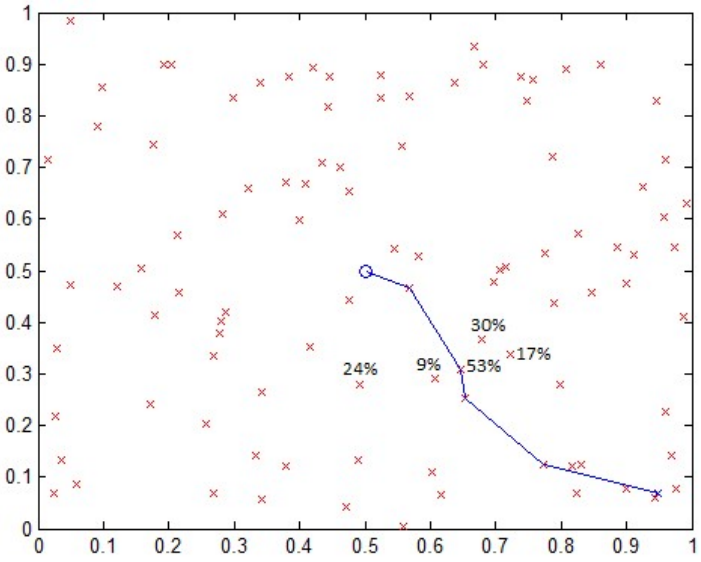


Figura 30 - Caminho percorrido pela informação de uma grandeza medida.

Nesta simulação foi observado que a informação passou por quatro nós do ambiente até chegar ao gateway. Observa-se também que o terceiro nó escolhido foi aquele que possuía maior nível de bateria entre os nós próximos.

Para verificar este comportamento, foi realizada uma nova simulação na qual foi mantido o mesmo sensor de origem, porém a bateria do terceiro nó percorrido no caminho anterior foi modificada. Inicialmente ela possuía o valor de 53% e nesta simulação ele foi alterado para um valor baixo, igual a 5%.

O novo caminho de transmissão obtido pode ser visto na Figura 31.

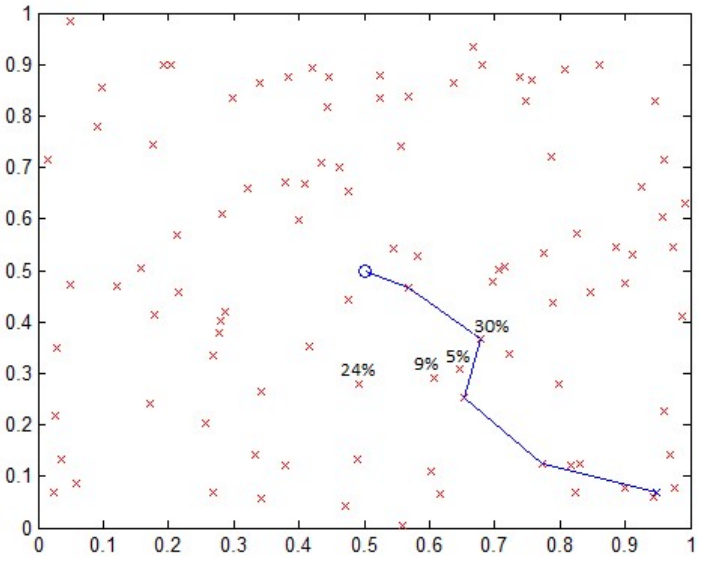


Figura 31 - Caminho percorrido pela informação de uma grandeza medida após variação da bateria.

Como a bateria do nó usado anteriormente está agora em um nível baixo, é preferível poupar este roteador para casos em que ele precise necessariamente ser utilizado. Observa-se que o novo caminho utilizado percorreu o nó que tinha o maior nível de bateria ainda aceitável, ou seja, 30%, e que também estava próximo do *gateway*.

Uma terceira simulação foi realizada alterando a bateria deste nó utilizado de 30% para 10%. O resultado obtido está na Figura 32.

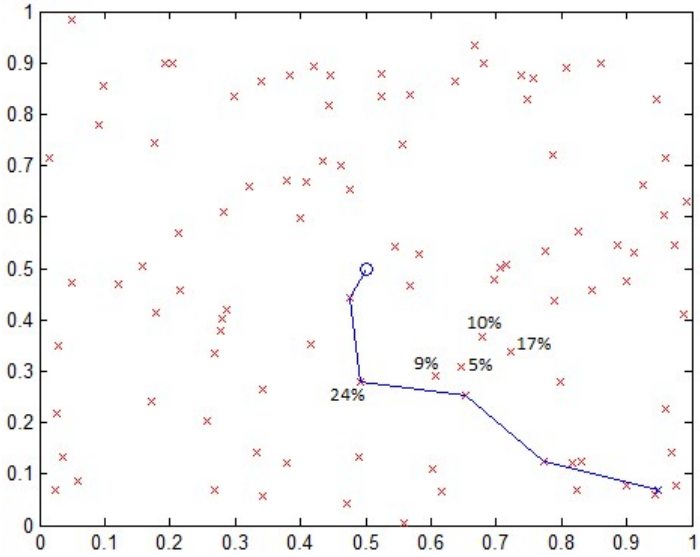


Figura 32 - Caminho percorrido pela informação de uma grandeza medida após segunda variação da bateria.

Neste caso, o nó mais próximo e com um nível de bateria superior, 24%, estava localizado mais à esquerda e foi escolhido em alternativa aos percursos encontrados anteriormente.

Em um sistema real, este processo seria executado simultaneamente em diversos nós de acordo com a necessidade da planta. Porém, a simulação não foi executada com mais sensores para possibilitar uma melhor visualização do comportamento em cada nó.

Através das simulações realizadas é possível perceber o comportamento adequado de avaliação através do sistema fuzzy implementado, que priorizou a escolha de um caminho mais curto até o *gateway*, porém considerou a quantidade de energia residual em cada roteador a fim de prolongar o tempo de vida útil da

rede. Assim, ficou descrito o processo para criação de um agregador fuzzy e o comportamento satisfatório da avaliação de dois objetivos (COELHO et al., 2017).

4.2 Maximização de funções

Como primeiro estudo de caso utilizando um algoritmo genético, foi executada uma aplicação com o objetivo de encontrar o ponto que maximiza duas funções multimodais simultaneamente (GONÇALVES et al., 2009).

Para isso, as duas funções utilizadas foram a função *f6* (8) e a função *f6* com um deslocamento na variável *x* (9).

$$f6 = 0{,}5 - \frac{\left(sen\sqrt{x^2 + y^2}\right)^2 - 0{,}5}{\left[1 + 0{,}001\left(x^2 + y^2\right)\right]^2} \tag{8}$$

$$f6_{desl} = 0{,}5 - \frac{\left(sen\sqrt{(x + 0{,}14)^2 + y^2}\right)^2 - 0{,}5}{\left[1 + 0{,}001\left((x + 0{,}14)^2 + y^2\right)\right]^2} \tag{9}$$

A Figura 33 ilustra as duas funções em um intervalo de [-10,10] para as variáveis *x* e *y*. A Figura 34 ilustra a composição das duas funções em duas dimensões fixando *y*=0. É possível observar que as funções possuem um deslocamento de 0,14 na variável *x* e o valor ótimo que maximiza as duas funções simultaneamente é $f6=f6_{desl}$=0,9951, localizado no ponto *x*=-0,07 e *y*=0. Observa-se também que as funções possuem diversos pontos bem próximos do valor ótimo global.

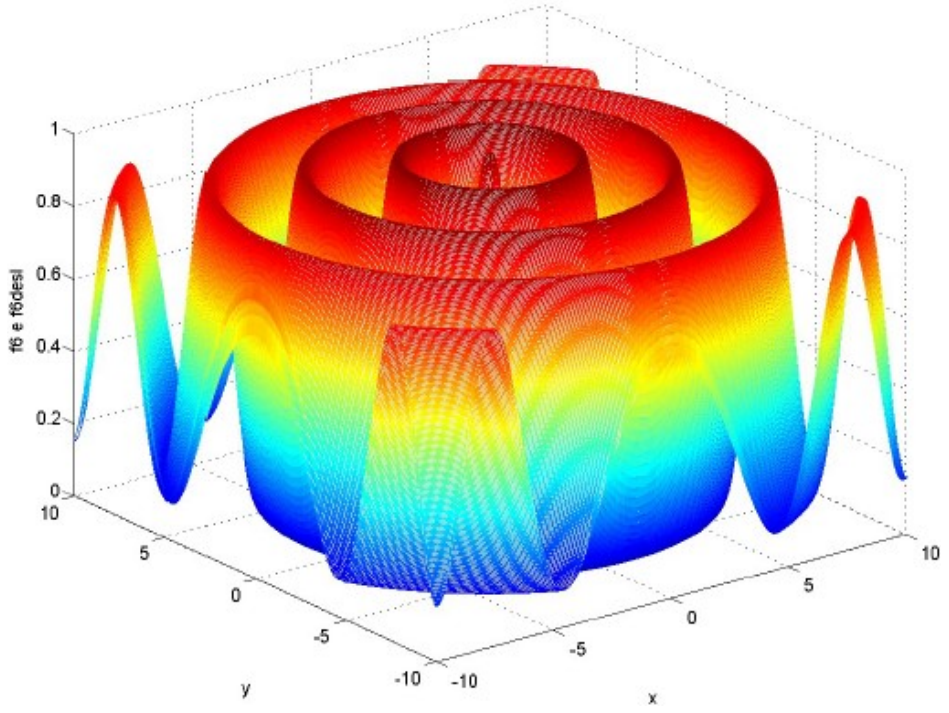


Figura 33 – Funções *f6* e $f6_{desl}$.

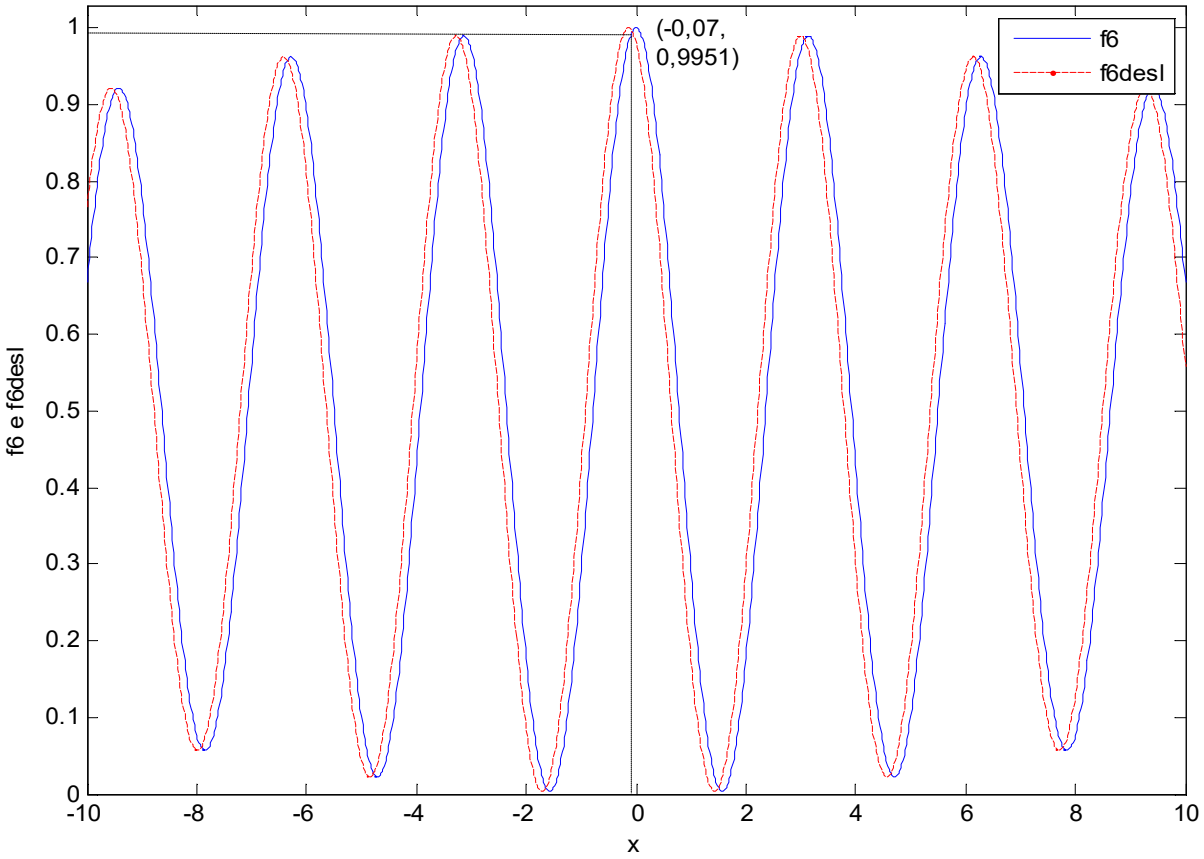


Figura 34 - Funções *f6* e $f6_{desl}$ em duas dimensões.

Neste trabalho foram utilizados os resultados apresentados em (GONÇALVES et al., 2009) para comparação com o método de agregação fuzzy, para isso, a execução do algoritmo genético foi realizada com os mesmos parâmetros de evolução utilizados por Gonçalves et al.. Um intervalo de busca de [-100,100] foi utilizado para as variáveis *x* e *y* e foram executadas 50 rodadas.

O estudo comparativo apresentado em (GONÇALVES et al., 2009) utilizou a técnica de agregação de objetivos, a técnica de Pareto ótimo e a técnica de minimização de energia (ZEBULUM, 1999). Para o método de agregação de objetivos, utilizaram-se pesos iguais e fixos para os dois objetivos, $w_1 = 0,5$ e $w_2 = 0,5$. Para o método de minimização de energia, deve ser definida a constante de balanceamento α, que neste caso é igual a 0,7, e os pesos iniciais que são $w_1 = 0,5$ e $w_2 = 0,5$ e atualizados durante a evolução. No método de Pareto, foi atribuída aos indivíduos não dominados uma aptidão igual a 100, acrescida do número de indivíduos dominados.

Neste trabalho, a avaliação do AG foi feita através do agregador fuzzy proposto. A composição das funções de pertinência e regras foi realizada de forma a inserir as especificações do problema e preferências do projetista e a mesma metodologia pode ser utilizada para aplicação em outros problemas de maximização.

Inicialmente, foram criadas as funções de pertinência representando as entradas. Como o objetivo é maximizar as duas funções simultaneamente, as funções de pertinência para os dois objetivos (duas funções) são iguais (Figura 35).

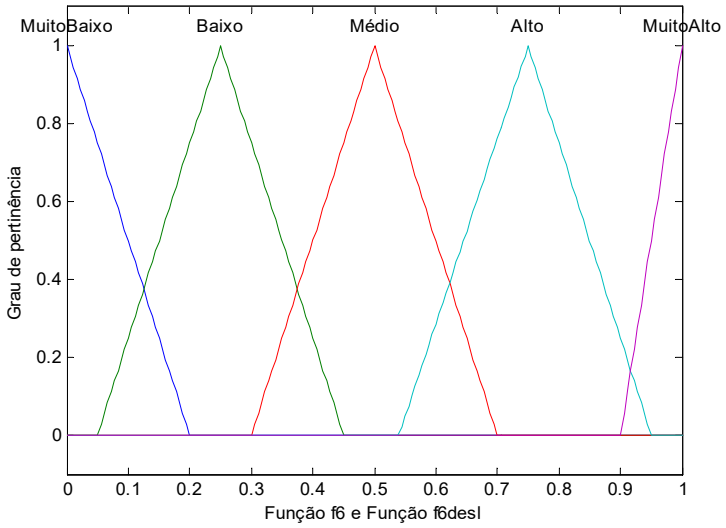


Figura 35 – Funções de pertinência das entradas.

O formato triangular foi utilizado por representar funções de simples manipulação e que geralmente atendem de forma adequada aos objetivos. Cinco funções de pertinência foram criadas e dispostas uniformemente dentro dos limites.

Cabe ressaltar que é necessário saber os limites da imagem das funções objetivo. Neste caso, a variação dos valores para *f6* e $f6_{desl}$ está dentro do intervalo de [0,1]. Para extensão deste modelo para outras funções é necessário usar valores normalizados neste intervalo, de forma a facilitar e generalizar a implementação, ou ajustar os valores para atender os limites da função.

Ajustes nos limites das funções de pertinência devem ser realizados de forma a adaptar e inserir as preferências do problema. Através de resultados experimentais foi realizado um ajuste no limite inferior da função de pertinência *Muito Alto* e no limite superior da função *Alto* para diferenciar melhor os valores que se encontram neste intervalo e melhorar o resultado.

Para as FP's da saída foi realizado o mesmo processo. Cinco FP's são utilizadas representando uma aptidão *Muito Ruim*, *Ruim*, *Média*, *Boa* e *Muito Boa* variando no intervalo de 0 a 10, conforme Figura 36.

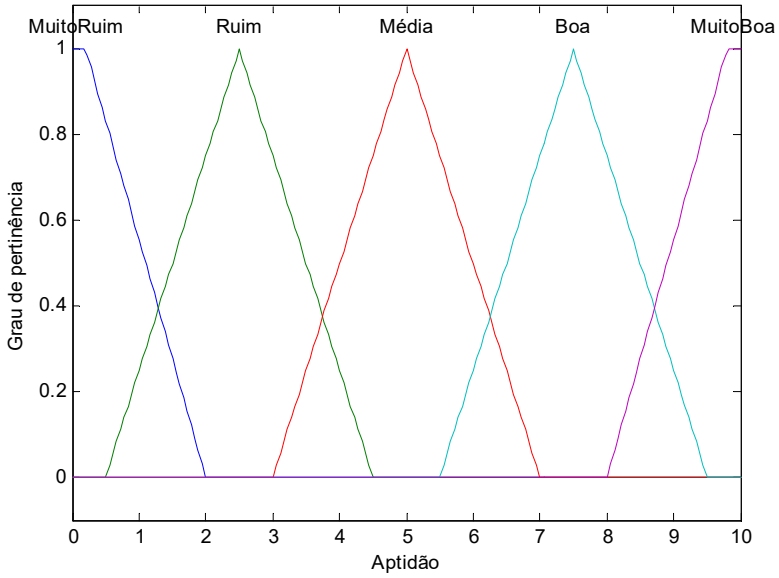


Figura 36 - Funções de pertinência da saída.

As regras elaboradas também atendem às duas funções igualmente e estão descritas na Tabela 3. A criação das regras foi realizada a partir da análise das FP's de acordo com o objetivo de maximização dos dois objetivos. Por exemplo, caso as

duas funções apresentem valores do termo linguístico *Muito Alto* a avaliação é *Muito Boa*. Da mesma forma, caso as duas funções apresentem valores *Muito Baixo* a avaliação é *Muito Ruim*. Para casos em que uma função apresenta valor *Médio* e a outra *Muito Alto* a avaliação é *Boa*.

Tabela 3 – Regras para maximização de funções.

Função 1 / **Função 2**	*Muito Baixo*	*Baixo*	*Médio*	*Alto*	*Muito Alto*
Muito Baixo	Muito Ruim	Muito Ruim	Muito Ruim	Ruim	Ruim
Baixo	Muito Ruim	Muito Ruim	Ruim	Média	Média
Médio	Muito Ruim	Ruim	Média	Boa	Boa
Alto	Ruim	Média	Boa	Boa	Muito Boa
Muito Alto	Ruim	Média	Boa	Muito Boa	Muito Boa

Em Gonçalves et al. (2009) são apresentados os melhores resultados encontrados entre as 50 rodadas para o método de agregação de objetivos, o método de Pareto ótimo e o método de minimização de energia. Os resultados obtidos para estes três métodos são apresentados na Tabela 4 juntamente com o melhor resultado obtido pelo método de agregação fuzzy proposto.

Tabela 4 – Comparação entre os melhores resultados de cada método.

	Agregação de Objetivos	*Pareto Ótimo*	*Minimização de Energia*	*Agregação Fuzzy*
Melhor valor de x	-0,0726462	-0,0665903	-0,070262	**-0,0700537**
Melhor valor de y	**-0,0000238**	0,0045538	**0,0000238**	0,0023123
Melhor valor de f6	0,9947270	0,9955470	0,9950660	**0,9950903**
Melhor valor de $f6_{desl}$	0,9924660	0,9945950	0,9951400	**0,9951053**

A partir dos resultados obtidos observa-se que o método da minimização de energia e o método de agregação fuzzy são os métodos que mais se aproximaram do resultado esperado para as funções objetivo. Os resultados dos outros métodos também foram satisfatórios, bem próximos um do outro e da solução ideal em ambos os objetivos. Assim, os resultados não possuem diferença significativa e não é possível afirmar a superioridade de um método em relação a outro.

No entanto, o teste realizado comprova que o método de agregação fuzzy avalia de forma adequada o AG, atendendo de forma satisfatória aos dois objetivos estabelecidos.

4.3 Minimização de funções – Problemas de Benchmark

Existem na literatura alguns problemas considerados referência para validação de técnicas multiobjetivo (HUBAND et al., 2006).

Com a finalidade de comparar os resultados da técnica apresentada neste trabalho aos resultados obtidos por técnicas tradicionais de otimização multiobjetivo, foi escolhido utilizar três dos problemas teste propostos por Zitzler, Deb e Thiele, denominados ZDT1, ZDT2 e ZDT3 (ZITZLER et al., 2000). Tais problemas foram selecionados pelo fato de possuírem grande uso em trabalhos relacionados (CHENG et al., 2016) (JARIYATANTIWAIT, 2015) e por possibilitarem uma boa visualização da fronteira de Pareto, viabilizando uma comparação adequada.

Os problemas teste são compostos por duas funções objetivo, f_1 e f_2, com o intuito de minimizá-las. Tais problemas podem ser matematicamente definidos conforme a seguir:

Dado	$x = \{x_1, ... x_k\}$
Onde	$y = \{x_1\}$
	$z = \{x_2, ..., x_k\}$
Minimizar	$f_1(y)$
e	$f_2(y, z) = g(z)h(f_1(y), g(z))$

O objetivo dos problemas é encontrar o vetor de variáveis $x=\{x_1,..x_n\}$ que minimiza as duas funções objetivo f_1 e f_2. As funções objetivo correspondentes a cada um dos problemas são apresentadas na Tabela 5 e todas possuem o domínio das variáveis entre [0,1].

Todos os problemas ZDT envolvem funções com objetivos conflitantes, pois a minimização de uma função implica em valores mais altos para a outra função.

Tabela 5 – Problemas multiobjetivo ZDT.

PROBLEMA	***FUNÇÕES***
ZDT1	$f_1 = y_1$ $g = 1 + 9\sum_{i=1}^{k-1} \frac{z_i}{k-1}$ $h = 1 - \sqrt{\frac{f_1}{g}}$ $f_2 = g.h$
ZDT2	$f_1 = y_1$ $g = 1 + 9\sum_{i=1}^{k-1} \frac{z_i}{k-1}$ $h = 1 - \left(\frac{f_1}{g}\right)^2$ $f_2 = g.h$
ZDT3	$f_1 = y_1$ $g = 1 + 9\sum_{i=1}^{k-1} \frac{z_i}{k-1}$ $h = 1 - \sqrt{\frac{f_1}{g}} - \left(\frac{f_1}{g}\right) sen(10\pi f_1)$ $f_2 = g.h$

Para analisar os resultados obtidos pelo método de agregação fuzzy, foram geradas as fronteiras de Pareto através do algoritmo SPEA2. Este algoritmo foi escolhido por apresentar bons resultados na literatura. Os códigos usados estão disponíveis em *www.yarpiz.com*.

Como o método de agregação fuzzy resulta em uma única solução e não encontra a fronteira completa de Pareto, foi analisado se a solução obtida pertence à fronteira de Pareto gerada pelo método tradicional.

Cada função objetivo (f_1 e f_2) corresponde a uma das entradas do agregador fuzzy. As funções de pertinência implementadas para as entradas e para a saída são mostradas na Figura 37 e na Figura 38, respectivamente. Foram usadas cinco funções de pertinência com formato triangular para cada entrada e saída. As duas entradas possuem funções de pertinência iguais e os conjuntos foram distribuídos uniformemente dentro do intervalo de [0,1]. A primeira entrada, correspondente ao primeiro objetivo, varia, de fato, dentro do intervalo entre *0* e *1*, porém a segunda entrada pode alcançar valores maiores que *1* e menores que *0* dependendo de qual dos três problemas é abordado. Desta forma, para facilitar a implementação, os valores aplicados ao sistema devem ser normalizados neste intervalo.

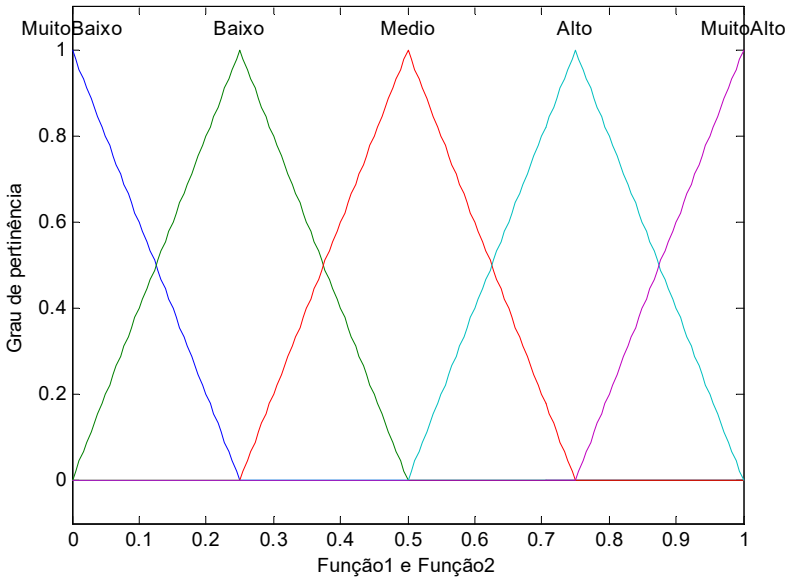


Figura 37 – Funções de pertinência das funções de entrada dos problemas ZDT.

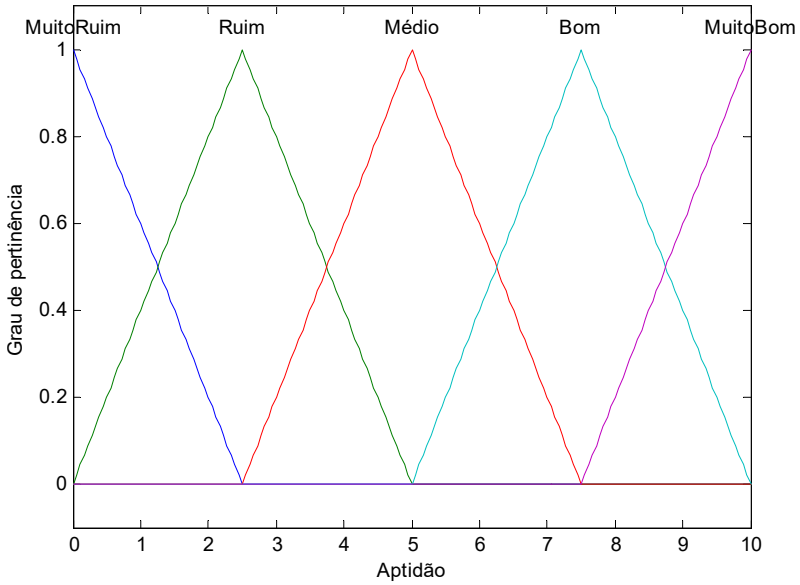


Figura 38 – Funções de pertinência para Aptidão dos problemas ZDT.

As regras foram criadas de forma a minimizar os dois objetivos. Por exemplo, soluções correspondentes ao termo linguístico *Muito Baixo*, para os dois objetivos, recebem avaliação *Muito Bom*, enquanto soluções correspondentes a *Muito Alto* recebem uma avaliação *Muito Ruim*. Todas as regras são mostradas na Tabela 6.

Tabela 6 – Regras do agregador fuzzy para problemas ZDT.

Função 1 / **Função 2**	*Muito Baixo*	*Baixo*	*Médio*	*Alto*	*Muito Alto*
Muito Baixo	Muito Bom	Muito Bom	Bom	Médio	Ruim
Baixo	Muito Bom	Bom	Médio	Médio	Ruim
Médio	Bom	Médio	Médio	Ruim	Muito Ruim
Alto	Médio	Médio	Ruim	Muito Ruim	Muito Ruim
Muito Alto	Ruim	Ruim	Muito Ruim	Muito Ruim	Muito Ruim

Alterações nas regras podem ser realizadas quando há o intuito de alterar a preferência de minimização das funções objetivo.

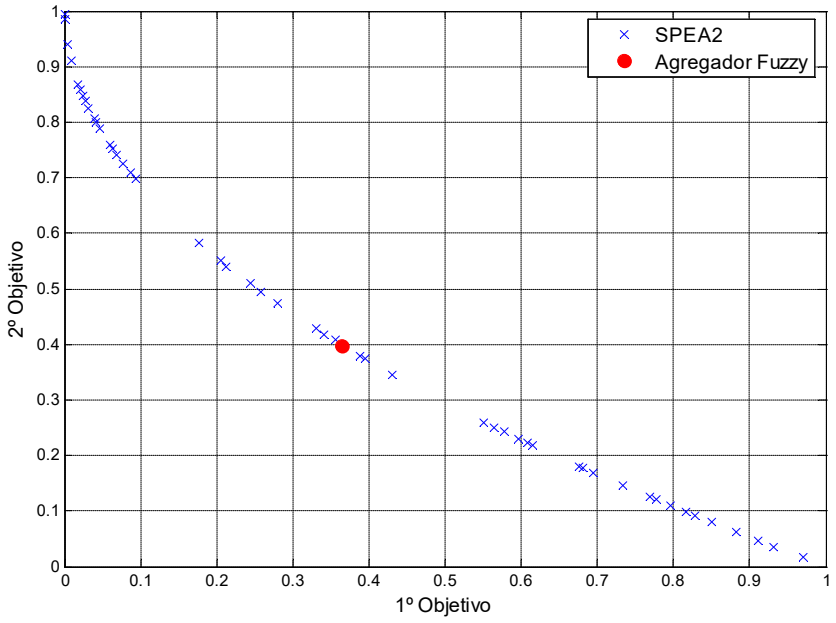
Para o algoritmo genético, foram atribuídos os parâmetros mostrados na Tabela 7.

Tabela 7 – Configuração do AG para ZDT.

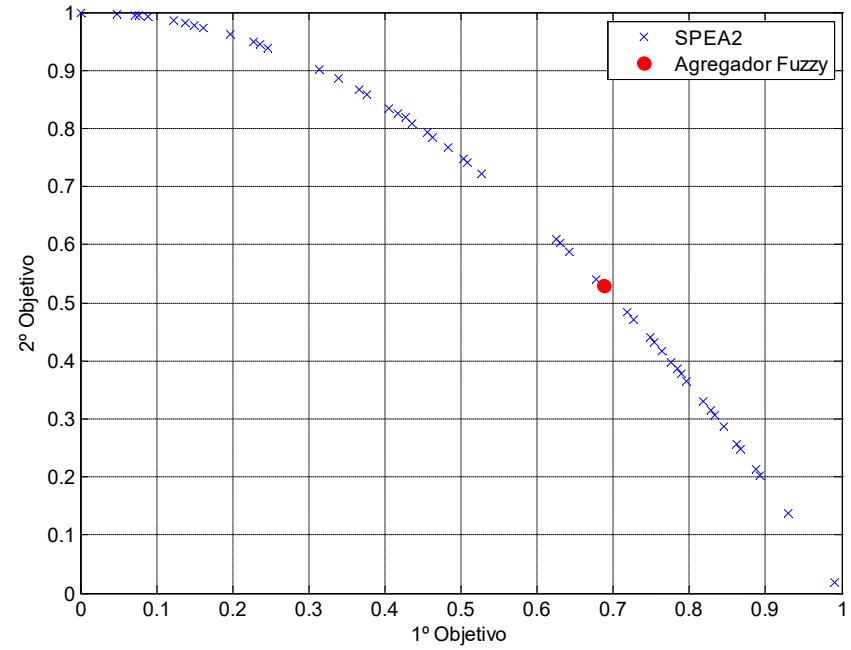
Parâmetro	*Valor*
Tamanho da população	100
Número de gerações	100
Taxa de crossover	90%
Taxa de mutação	5%

O número de variáveis da função, ou seja, o tamanho desejado do vetor *x*, deve estar compreendido entre 2 e 30. Neste trabalho são apresentados os testes realizados utilizando duas e cinco variáveis.

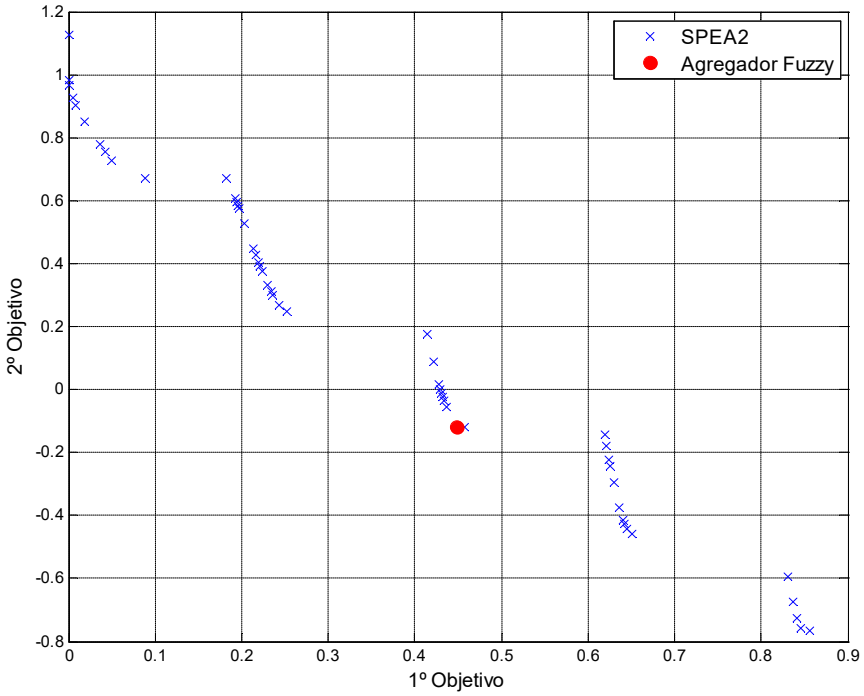
A Figura 39 mostra os resultados obtidos para o espaço objetivo dos três problemas, utilizando duas variáveis. O eixo *x* apresenta o 1º objetivo (f_1) e o eixo *y* o 2º objetivo (f_2). Os pontos identificados por um '*x*' azul são os pontos não dominados da fronteira de Pareto encontrados pelo algoritmo SPEA2. O círculo em vermelho é o ponto encontrado pelo agregador fuzzy.



(a) ZDT1



(b) ZDT2



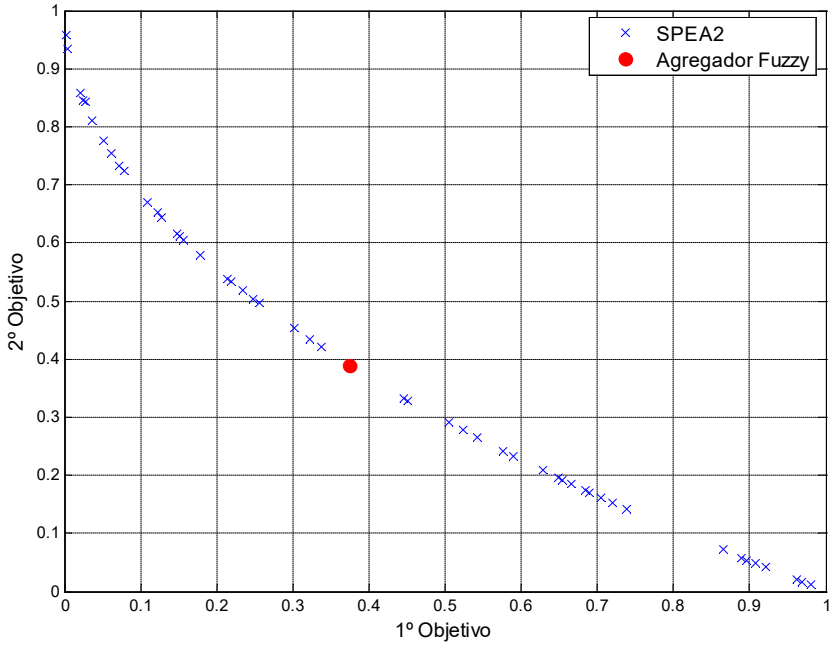
(c) ZDT3

Figura 39 – Resultados obtidos para três problemas ZDT com 2 variáveis.

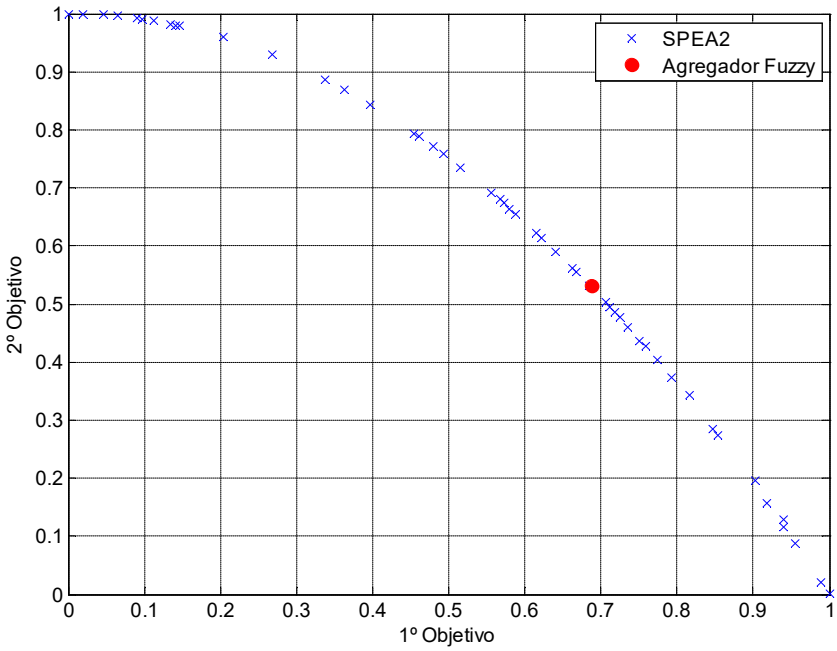
O problema ZDT1 apresenta a fronteira ótima de Pareto com formato convexo com os dois objetivos variando entre 0 e 1. A fronteira de Pareto do problema ZDT2 possui um formato côncavo que varia também entre 0 e 1. O problema ZDT3 é mais complexo e apresenta uma fronteira com formato descontínuo.

Observa-se através da Figura 39 que o agregador fuzzy alcançou nos três casos respostas localizadas na fronteira de Pareto obtida pelo algoritmo SPEA2 e próximos do ponto ideal que minimizaria as duas funções.

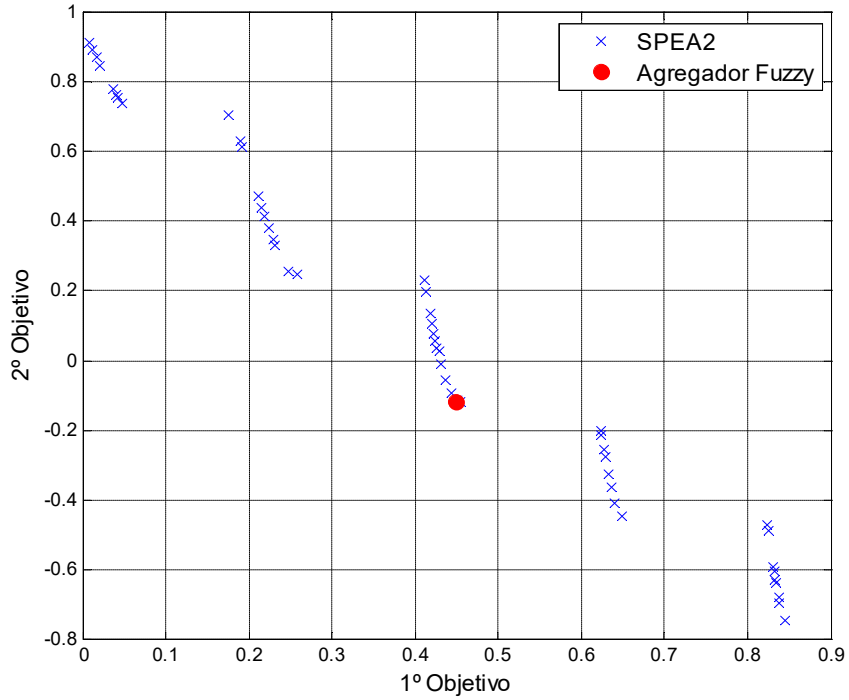
Da mesma forma, a Figura 40 mostra os resultados para funções com cinco variáveis.



(a) ZDT1



(b) ZDT2



(c) ZDT3

Figura 40 - Resultados obtidos para três problemas ZDT com 5 variáveis.

Para os problemas com 5 variáveis foram necessárias mais gerações (500). Esta necessidade provavelmente acontece devido ao aumento do número de variáveis que resulta na ampliação do espaço de busca dificultando o processo de evolução.

Apesar disso, os resultados encontrados são satisfatórios e demonstram que o método de avaliação utilizado atende aos objetivos requisitados de forma equivalente a um método tradicionalmente utilizado em problemas multiobjetivo, como o SPEA2, obtendo bons resultados que estão localizados na fronteira de Pareto.

4.4 Síntese de sistemas fuzzy

O algoritmo de aprendizagem da base de conhecimento de sistemas fuzzy utilizado neste trabalho é baseado em (AMARAL, 2003) e envolve duas etapas:

- Geração da base de conhecimento: responsável por gerar sistemas fuzzy com diferentes representações semânticas e suas respectivas bases de regras.
- Otimização das funções de pertinência: responsável pela busca dos parâmetros mais adequados para o melhor indivíduo selecionado na etapa anterior objetivando melhorar o seu desempenho.

As duas etapas utilizam algoritmos genéticos, porém elas são independentes e é possível executar apenas uma delas separadamente de acordo com o objetivo do projeto.

Na primeira etapa, cada cromossomo representa um conjunto completo de regras, sendo que cada gene corresponde ao consequente de uma regra. O tamanho do cromossomo é definido pelo número total de possíveis regras. A Figura 41 ilustra o modelo de representação descrito.

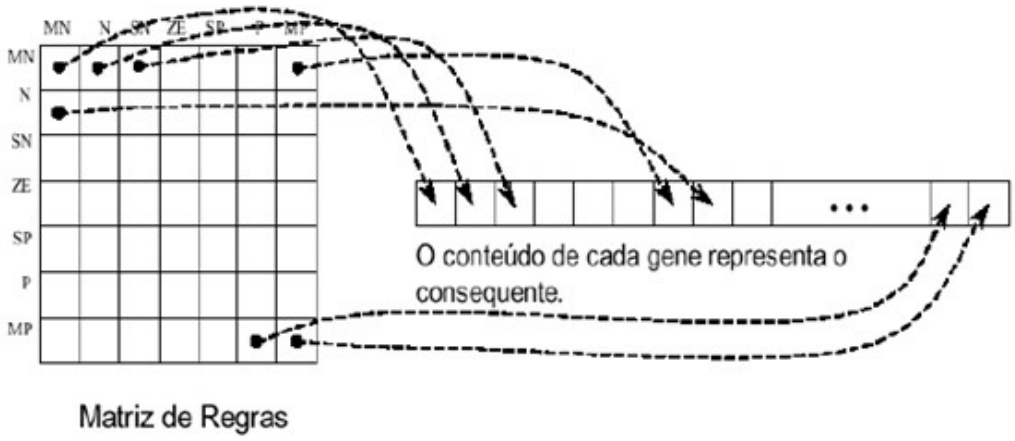


Figura 41 – Representação do cromossomo para obtenção de regras (AMARAL, 2003).

No sistema fuzzy apresentado na Figura 41 existem duas entradas com sete FP's cada. A matriz de regras contém 49 possíveis regras que são representadas sequencialmente em um vetor. Os valores de cada posição do vetor variam de acordo com o número de conjuntos da variável de saída.

O cromossomo possui tamanho variável, assim, o mecanismo do algoritmo é capaz de acrescentar novas funções de pertinência ou retirar funções de cada variável a fim de obter o melhor desempenho. Ao final da execução o sistema fuzzy com melhor avaliação é salvo.

Na segunda etapa, o algoritmo busca a sintonia das funções de pertinência. Os genes do cromossomo correspondem aos parâmetros das funções de pertinência de entrada e saída, por exemplo, os vértices para conjuntos triangulares, ordenados sequencialmente.

A obtenção da base de regras de sistemas fuzzy e ajuste das funções de pertinência foi aplicada a dois bancos de dados disponíveis em *www.keel.es* (ALCALÁ-FDEZ et al., 2011) e comparada a resultados encontrados na literatura.

Em ambos os problemas dois objetivos foram analisados na etapa de obtenção da base de regras:

- Acurácia;
- Interpretabilidade.

Na segunda etapa, correspondente à sintonia das funções, foi avaliada apenas a acurácia como objetivo, pois foram inseridas restrições no algoritmo para limitar a variação das funções de pertinência, garantindo a interpretabilidade.

Nas duas etapas a avaliação da acurácia é realizada através do *MSE* (*Mean Squared Error*):

$$MSE = \frac{1}{2D}\sum_{i=i}^{D}\left(F\left(x^i\right) - y^i\right)^2. \tag{10}$$

Onde *D* é o número total de amostras, $F\left(x^i\right)$ é a saída obtida para a amostra i e y^i é a saída esperada para a amostra i.

A equação (10) representa de fato o $MSE\ /2$, mas, assim como nos trabalhos de referência, será adotada a nomenclatura *MSE*, pois é a mais utilizada no campo de SFG's.

A interpretabilidade é avaliada conforme o número de regras obtidas. O valor é normalizado de acordo com o total de regras possíveis.

A Figura 42 ilustra o agregador fuzzy proposto de forma a avaliar a obtenção de regras com base nos dois objetivos.

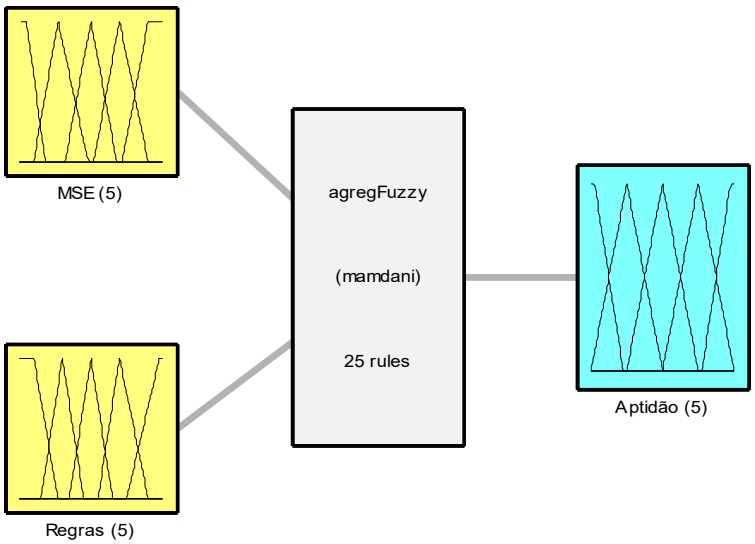
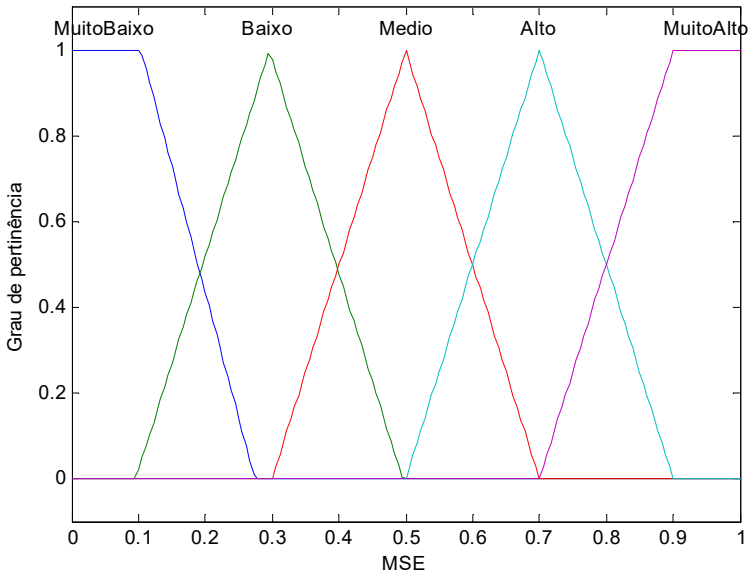


Figura 42 – Agregador fuzzy para obtenção de sistemas fuzzy.

As funções de pertinência para as entradas e saída seguem o formato mostrado na Figura 43. As entradas são normalizadas, a fim de atender a diferentes problemas e são usadas cinco FP's triangulares e trapezoidais para as entradas e saídas.



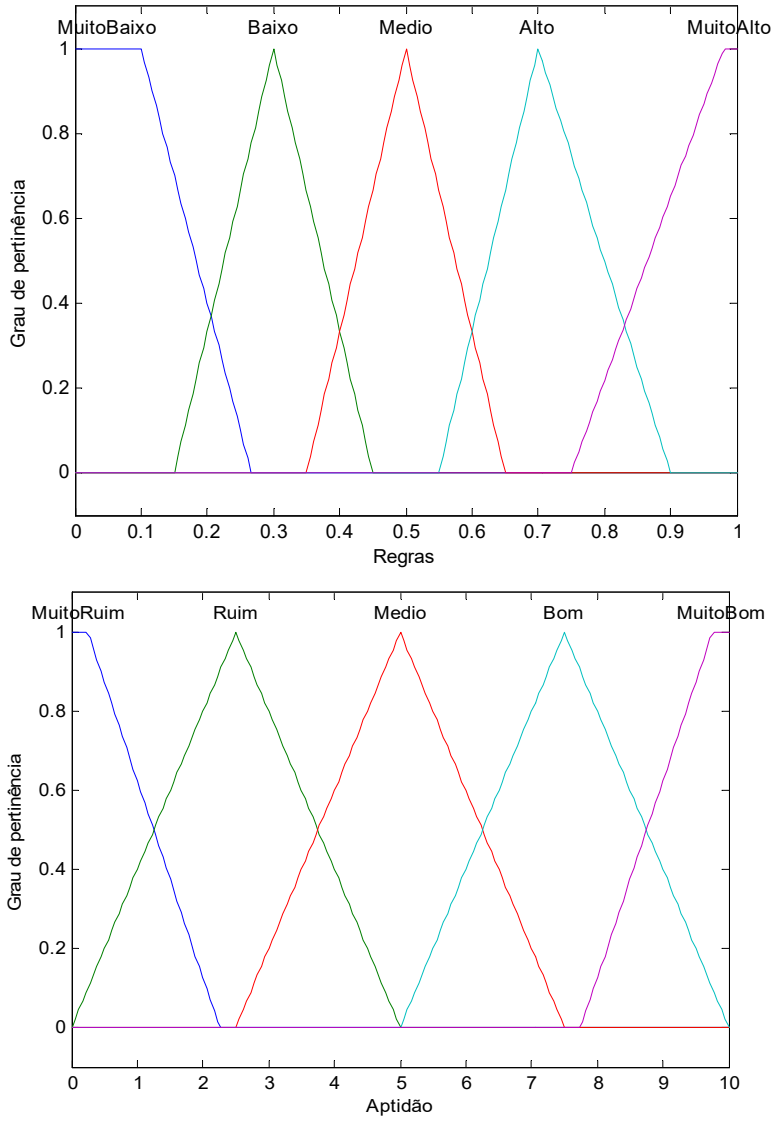


Figura 43 – Funções de pertinência para obtenção de sistemas fuzzy.

As regras do agregador fuzzy foram formuladas de forma a priorizar soluções com o *MSE* menor, porém a aptidão diminui à medida que o número de regras aumenta. Desta forma busca-se alcançar soluções com *MSE* e número de regras baixo. As regras utilizadas são apresentadas na Tabela 8.

Tabela 8 – Regras para síntese de sistemas fuzzy.

Regras / **MSE**	*Muito Baixo*	*Baixo*	*Médio*	*Alto*	*Muito Alto*
Muito Baixo	Muito Bom	Muito Bom	Muito Bom	Muito Bom	Bom
Baixo	Muito Bom	Muito Bom	Bom	Bom	Bom
Médio	Bom	Bom	Médio	Médio	Ruim
Alto	Médio	Médio	Médio	Ruim	Ruim
Muito Alto	Ruim	Ruim	Muito Ruim	Muito Ruim	Muito Ruim

4.4.1 Sistema Fuzzy 1: Problema *Plastic*

O conjunto de dados *Plastic* (*www.keel.es*) é aplicado a um problema de regressão em que o objetivo é determinar a pressão que uma determinada peça de plástico suporta quando aplicada determinada força a uma temperatura fixa.

No presente trabalho o objetivo é encontrar um sistema fuzzy para aplicação neste problema, obtendo a melhor acurácia e melhor interpretabilidade, através da execução de um AG multiobjetivo para obtenção das regras e ajuste dos parâmetros. O agregador fuzzy apresentado na seção anterior é utilizado para avaliar a evolução.

O banco de dados contém 160 amostras. Para o processo de avaliação da evolução foi utilizada a técnica de validação cruzada *k-fold* com cinco partições. Assim, em cada execução, 80% dos dados foram utilizados para o treinamento e 20% para o teste. O algoritmo foi executado seis vezes para cada partição de teste, totalizando trinta execuções.

A Tabela 9 apresenta as variáveis de entrada e saída do sistema fuzzy e os limites correspondentes a cada uma.

Tabela 9 – Variáveis do Sistema Fuzzy do problema *Plastic*.

Descrição	*Tipo de variável*	*Limites*
Força	Entrada	[12,4 , 48,7]
Temperatura	Entrada	[200 , 300]
Pressão	Saída	[10 , 20]

A Tabela 10 apresenta os valores usados para o AG de geração de regras e o AG para sintonia. Estes valores foram obtidos através de testes realizados.

Tabela 10 – Configuração do AG para o problema *Plastic*.

Parâmetro	*AG para Regras*	*AG para Sintonia*
Tamanho da população	100	100
Número de gerações	100	100
Taxa de *crossover*	50%	90%
Taxa de mutação	0,08%	5%

Com os resultados obtidos na execução das 30 rodadas foi calculada a média do *MSE* de treinamento e de teste e do número de regras. Os resultados de quatro diferentes técnicas existentes na literatura foram usados para comparação: o TS e $TS_{SP2\text{-}SI}$ (GACTO et al., 2010) e LEL-TSK e MOEA-TSK (GACTO et al., 2013). O agregador fuzzy utilizou no processo de evolução sete funções de pertinência para cada variável de entrada, totalizado 49 possíveis regras. A Tabela 11 apresenta os resultados para cada técnica.

Tabela 11 – Comparação de resultados para o problema *Plastic*.

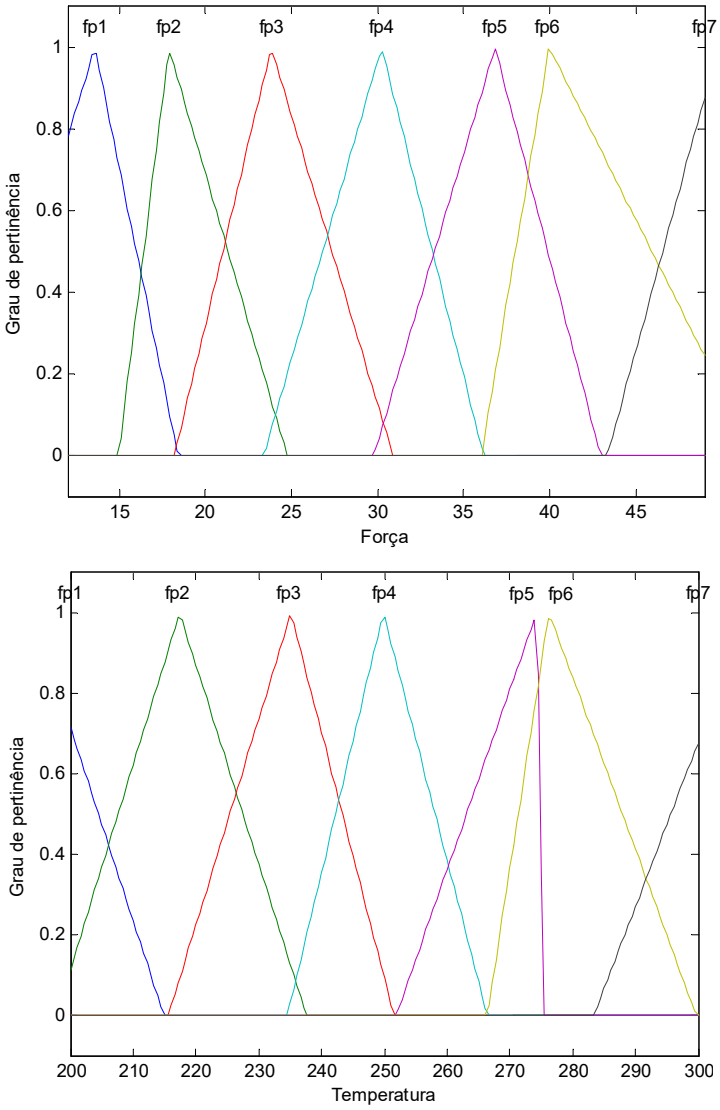
Sistema	*Método*	*MSE treinamento*	*MSE teste*	*Regras (nº médio)*
TSK	LEL -TSK	**1,032**	1,188	66
	MOEA-TSK	1,057	**1,136**	19,2
Mamdani	$TS_{SP2\text{-}SI}$	1,170	1,227	13,7
	Agregador Fuzzy	1,164	1,257	36,5
	TS	1,255	1,301	**13,3**

De acordo com os resultados mostrados da Tabela 13 o método LEL-TSK obteve o menor MSE de treinamento e o MOEA-TSK o menor erro de teste. Porém

cabe ressaltar que ambos correspondem a sistemas fuzzy do tipo Takagi-Sugeno-Kang (TSK) que é um modelo originalmente menos interpretável que o Mamdani.

Comparando o agregador fuzzy aos métodos TS e $TS_{SP2\text{-}SI}$ que também utilizam o modelo Mamdani, ele obteve um MSE de treinamento médio menor e um MSE de teste entre os dois. Por outro lado, o número de regras obtidas pelo agregador fuzzy foi maior.

As funções de pertinência mostradas na Figura 44 correspondem ao melhor sistema fuzzy encontrado durante as 30 rodadas. Tal sistema resultou em um MSE de treinamento igual a 1,1216. O sistema obteve ainda o segundo menor MSE de teste, igual a 1,1977 e 36 regras, que são mostradas na Tabela 12.



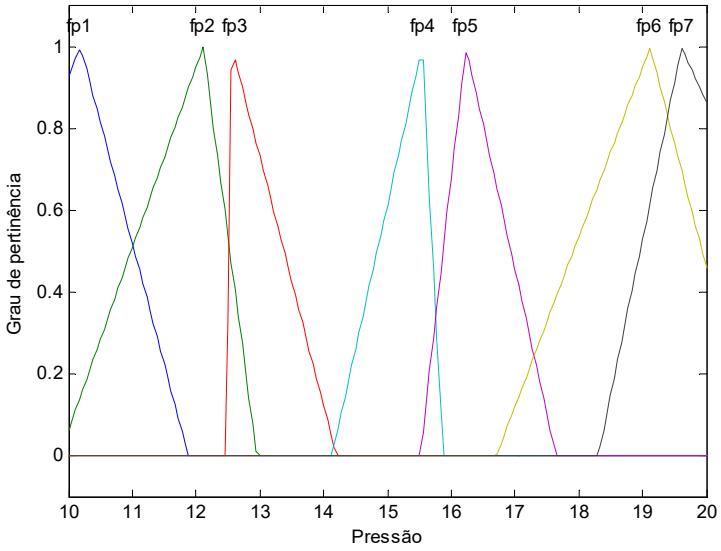


Figura 44 – Funções de pertinência do melhor resultado para o problema *Plastic*.

Tabela 12 – Regras do melhor resultado para o problema *Plastic*.

Temperatura / **Força**	*Fp1*	*Fp2*	*Fp3*	*Fp4*	*Fp5*	*Fp6*	*Fp7*
Fp1	Fp7	Fp5	Fp1	Fp1	-	Fp7	Fp1
Fp2	Fp7	Fp7	Fp3	Fp1	-	Fp4	Fp2
Fp3	-	Fp7	Fp7	Fp1	Fp1	-	Fp6
Fp4	Fp6	Fp7	Fp7	Fp4	Fp1	Fp1	-
Fp5	-	Fp6	Fp7	Fp7	Fp2	Fp1	-
Fp6	Fp7	-	Fp7	Fp7	Fp4	Fp1	-
Fp7	-	-	-	Fp7	Fp6	Fp1	-

4.4.2 Sistema Fuzzy 2: Problema *ELE1*

O problema denominado *ELE1* (CORDÓN et al., 1999) envolve a estimativa do comprimento de linhas de transmissão de baixa tensão em áreas rurais.

Para empresas de energia elétrica é interessante medir os custos de manutenção das suas próprias linhas de energia. No entanto, em alguns casos esses custos não podem ser calculados diretamente, pois a medição do

comprimento total das linhas de baixa tensão é um processo complicado e caro para realização em pequenas aldeias e núcleos rurais. Portanto, o comprimento deve ser estimado por meio de modelos indiretos.

Esta aplicação tem o objetivo de obter uma estimativa do comprimento total da linha de baixa tensão em um núcleo rural a partir de uma relação entre o número de habitantes do núcleo e a média da distância entre o centro do local e os três clientes mais distantes. Este é um problema que apresenta não linearidades que tornam a superfície do modelo complexa.

O objetivo neste trabalho é encontrar um sistema fuzzy capaz de realizar a estimação, obtendo a melhor relação entre acurácia e interpretabilidade. O sistema é evoluído através de um algoritmo genético avaliado pelo mesmo agregador fuzzy da aplicação anterior.

Para execução da evolução são usadas amostras reais de 495 cidades que, assim como no exemplo anterior, foram divididas em cinco partições e foi utilizada a técnica de validação cruzada. O algoritmo foi executado seis vezes para cada partição de teste, resultando em trinta rodadas no total.

A Tabela 13 apresenta as variáveis de entrada e saída do sistema fuzzy e os limites correspondentes.

Tabela 13 – Variáveis do Sistema Fuzzy do problema *ELE1*.

Descrição	*Tipo de variável*	*Limites*
Número de habitantes	Entrada	[1 , 320]
Raio médio do núcleo rural	Entrada	[60 , 1673,33]
Comprimento da linha de baixa tensão	Saída	[80 , 7675]

Os parâmetros de configuração do AG foram obtidos através de testes. A Tabela 14 apresenta os valores usados para o AG de geração de regras e o AG para sintonia.

Tabela 14 – Configuração do AG para o problema *ELE1*.

Parâmetro	*AG para Regras*	*AG para Sintonia*
Tamanho da população	100	100
Número de gerações	100	50
Taxa de crossover	80%	80%
Taxa de mutação	0,08%	5%

Com os resultados obtidos na execução das 30 rodadas foi calculada a média do *MSE* de treinamento, *MSE* de teste e do número de regras. Os resultados de quatro técnicas abordadas em (ALCALÁ et al., 2009) foram usados para comparação: o NSGA-II$_{RB}$ e o PAES$_{RB,}$ que buscam a base de regras e ajustes das funções, e o NSGA-II$_{KB}$ e o PAES$_{KB}$, que apenas ajustam a base de conhecimento, partindo de uma base pré-definida. Todas as quatro técnicas geram sistemas fuzzy do tipo Mamdani e o número de regras é definido para variar entre um mínimo de 5 regras e máximo de 30.

Em (ALCALÁ et al., 2009) são apresentados dados médios correspondentes a pontos específicos da fronteira de Pareto descritos a seguir e mostrados em um exemplo na Figura 45:

- Maior acurácia: é o ponto localizado mais à direita na fronteira de Pareto, correspondente a melhor acurácia, ou seja, menor MSE e maior número de regras.

- Relação média: é o ponto médio da fronteira que representa uma acurácia e número de regras médio.

- Menor complexidade: é o primeiro ponto da fronteira localizado na parte esquerda, em que o número de regras é mínimo, porém possui MSE maior.

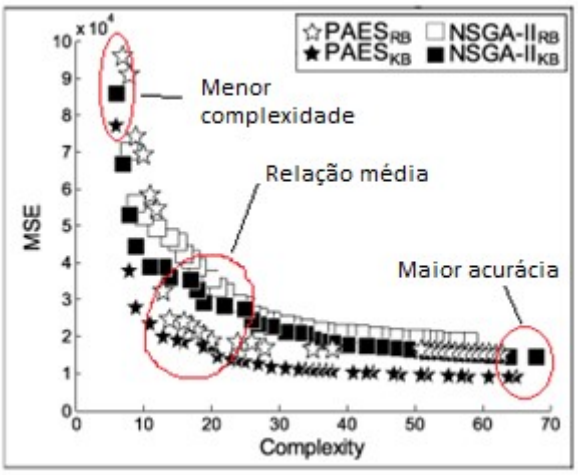


Figura 45 – Exemplo de localização dos pontos referentes às três situações analisadas (adaptado de ALCALÁ et al., 2009).

Os dados correspondentes a estas três situações encontram-se na Tabela 15, na Tabela 16 e na Tabela 17, respectivamente. Em todas as tabelas há também os valores encontrados para o agregador fuzzy, que gerou as regras a partir de 4 funções de pertinência para a primeira variável e 5 para a segunda, totalizando 20 regras possíveis.

Tabela 15 – Comparação de resultados para o problema *ELE1* – Maior acurácia.

Método	*MSE treinamento*	*MSE teste*	*Regras*
NSGA-II$_{RB}$	156952	205182	29
NSGA-II$_{KB}$	157613	202923	29
PAES$_{RB}$	156775	210162	29
PAES$_{KB}$	**145995**	194028	27
Agregador Fuzzy	170510	**184600**	**13**

Tabela 16 – Comparação de resultados para o problema *ELE1* – Relação média entre acurácia e regras.

Método	*MSE treinamento*	*MSE teste*	*Regras*
NSGA-II$_{RB}$	160988	204150	17
NSGA-II$_{KB}$	161089	204341	16
PAES$_{RB}$	159948	195961	17
PAES$_{KB}$	**148845**	189497	16
Agregador Fuzzy	170510	**184600**	**13**

Tabela 17 – Comparação de resultados para o problema *ELE1* – Menor número de regras.

Método	*MSE treinamento*	*MSE teste*	*Regras*
NSGA-II$_{RB}$	221761	245409	**5**
NSGA-II$_{KB}$	220192	244249	**5**
PAES$_{RB}$	214215	232268	**5**
PAES$_{KB}$	212337	234179	**5**
Agregador Fuzzy	**170510**	**184600**	13

Uma comparação entre os resultados das tabelas acima, mostra que o algoritmo PAES$_{KB}$ obteve o menor MSE de treinamento nas duas primeiras situações. Porém, o Agregador Fuzzy obteve o melhor MSE de teste nos três casos, o que demonstra que o sistema fuzzy evoluído conseguiu obter uma boa capacidade de generalização mantendo um número razoável de regras.

Além disso, o número de regras obtido pelo agregador fuzzy foi menor em relação aos quatro métodos nos dois primeiros casos. Em relação ao último caso, o número de regras encontrado foi maior, porém o MSE menor.

Dentre os 30 sistemas fuzzy encontrados durante a execução, o sistema com menor MSE de treinamento foi selecionado. Tal sistema obteve um MSE de treinamento igual a 156323 e MSE de teste igual a 176452 com 14 regras. As funções de pertinência correspondentes a este sistema são mostradas na Figura 46 e a Tabela 18 mostra as 14 regras obtidas.

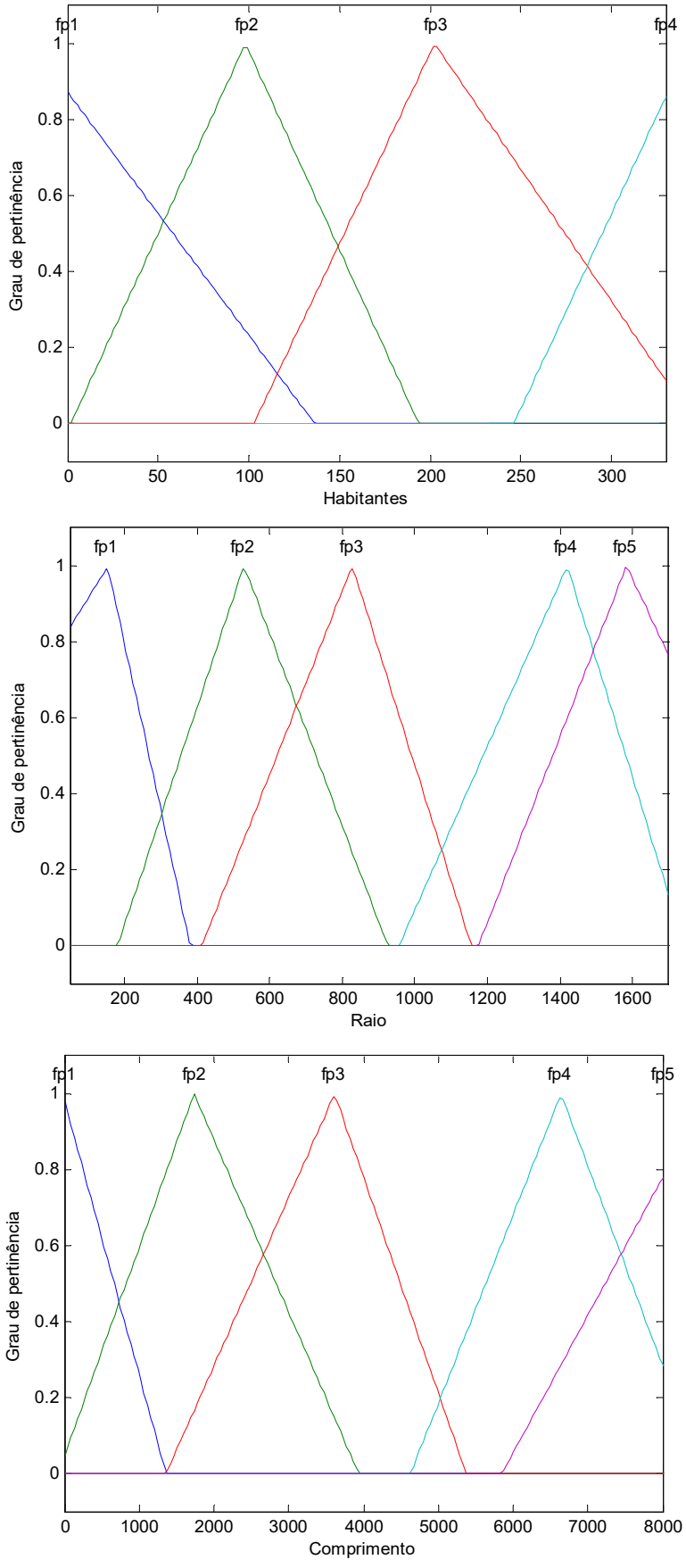


Figura 46 - Funções de pertinência do melhor resultado para o problema *ELE1.*

Tabela 18 – Regras do melhor resultado para o problema *ELE1.*

Habitantes / **Raio**	*Fp1*	*Fp2*	*Fp3*	*Fp4*
Fp1	Fp1	-	Fp1	Fp4
Fp2	Fp1	Fp2	Fp2	Fp2
Fp3	Fp2	Fp5	Fp4	-
Fp4	-	Fp3	-	-
Fp5	Fp2	Fp4	Fp2	-

Esses resultados mostram que o método utilizado foi satisfatório na geração de sistemas fuzzy, pois conseguiu obter funções de pertinência que podem ser diferenciadas, não estão sobrepostas ou incoerentes e obteve poucas regras quando comparado a outros estudos.

4.5 Sistemas de controle

Sistemas de controle são necessários em diversos campos de atuação. Obter um processo estável implica em resultados mais eficientes, produtos de melhor qualidade, diminuição do reprocessamento, economia de matéria prima dentre outros fatores altamente importantes, seja em uma indústria, laboratório ou qualquer ambiente que demanda um controle eficiente. Os controladores clássicos PID (Proporcional Integral Derivativo) possuem uma aplicabilidade geral na maioria dos sistemas de controle e correspondem a grande parte dos controladores industriais. Desta forma, um ajuste fino de seus parâmetros de controle é essencial para um processo estável.

Desde o aparecimento do primeiro método de sintonia para controladores, proposto por ZIEGLER & NICHOLS em 1942, várias técnicas de ajuste de PID têm sido propostas na literatura, entre elas, técnicas de controle inteligente, como a lógica fuzzy, redes neurais e algoritmos genéticos.

Baseando-se nesta necessidade, foram desenvolvidos neste trabalho alguns estudos aplicados à sintonia de controladores PID a fim de obter um desempenho adequado de controle. Para isso, utiliza-se a técnica de busca de um algoritmo genético para encontrar os melhores ganhos do controlador, ou seja, os ganhos proporcional, integral e derivativo (*Kp*, *Ki* e *Kd*).

A otimização multiobjetivo é aplicada neste problema com o intuito de obter o melhor sistema de acordo com cada necessidade de projeto.

O primeiro passo em projetos de sistemas de controle é a obtenção de um modelo matemático do sistema e a partir disso é possível analisar o seu desempenho.

Na análise do sistema de controle são utilizados sinais de entrada como referência que possibilitem uma comparação de desempenho baseando-se em determinadas especificações. Entre as principais especificações consideradas no domínio do tempo, destacam-se o *overshoot* ou máximo valor de ultrapassagem (M_p), o tempo de subida (t_r), o tempo de acomodação (t_s) e o tempo de atraso (t_d). A Figura 47 mostra esses quatro parâmetros.

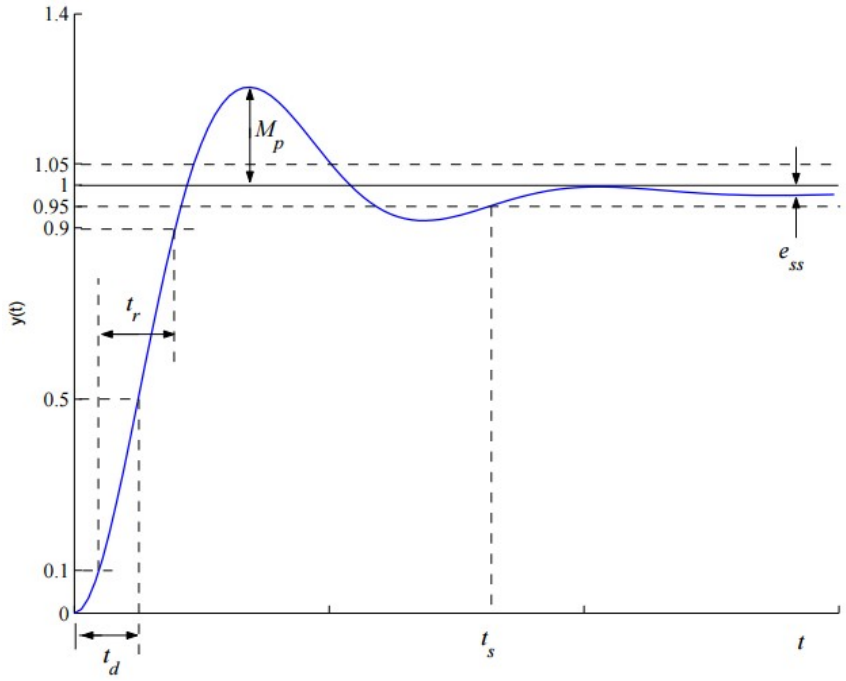


Figura 47 – Resposta de um sistema de controle.

O *overshoot* corresponde ao ponto máximo obtido além do sinal de referência. O tempo de subida é o tempo necessário para o sinal de saída variar de 10 a 90% do valor final. O tempo de acomodação corresponde ao tempo gasto para o valor acomodar a uma faixa (e_{ss}), geralmente 2% ou 5%, do valor final e o tempo de atraso é o tempo gasto para o sinal alcançar 50% do valor final.

Neste trabalho, optou-se por analisar três objetivos: o *overshoot*, o tempo de subida e o tempo de acomodação. Assim, o agregador fuzzy implementado possui três entradas e uma saída.

As funções de pertinência utilizadas para avaliação foram criadas a partir das especificações requeridas. Por exemplo, o *overshoot* é medido em porcentagem, logo, foi utilizada uma escala de 0 a 100% como mostrado na Figura 48. Para valores acima de 45% o *overshoot* é considerado *Muito Alto* e acima de 55% não é mais aceitável. Valores abaixo de 12% são considerados ideais e, portanto, caracterizam o termo linguístico *Baixo*. Valores em faixas intermediárias são considerados *Médio* ou *Alto*.

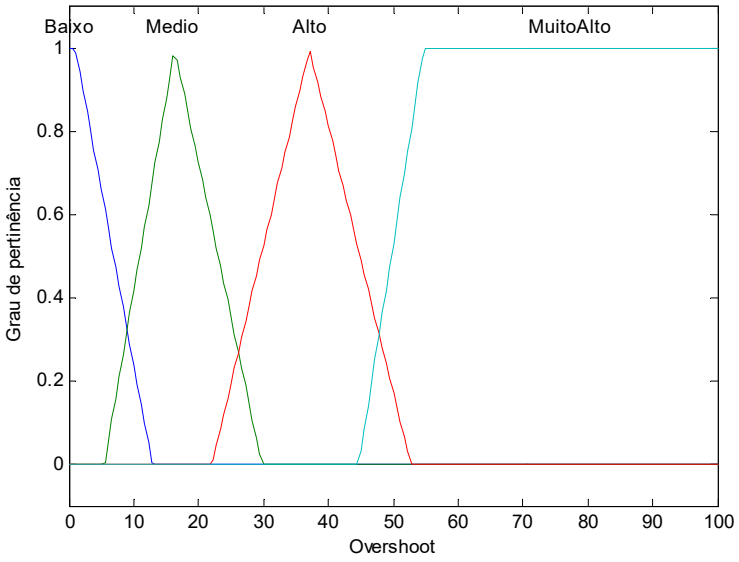


Figura 48 – Função de pertinência da variável de entrada Overshoot.

Para o tempo de subida foi considerado que um tempo acima de 1 segundo começa a ser *Muito Alto*, assim como, um tempo abaixo de 0,4 segundo é *Baixo* e desejável conforme ilustrado na Figura 49.

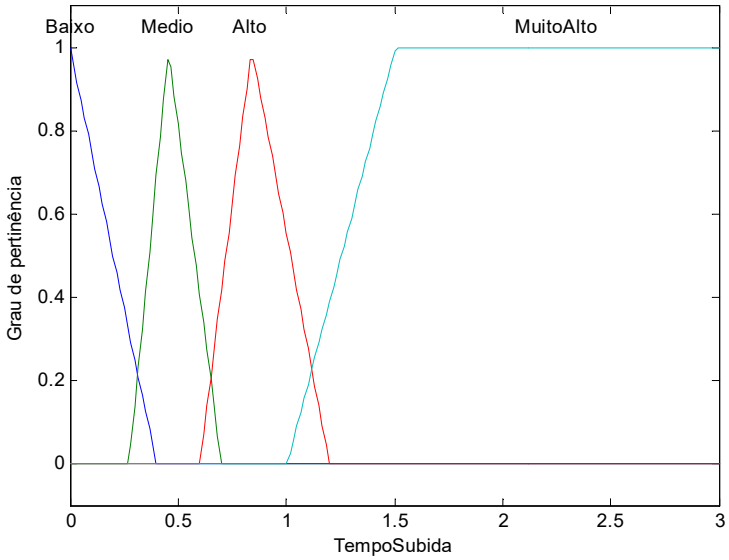


Figura 49 - Função de pertinência da variável de entrada Tempo de Subida.

O tempo de acomodação foi representado por três conjuntos *Baixo, Médio* e *Alto*, considerando o conjunto *Médio* centrado em 2 segundos, o *Baixo* para valores até 1,5 segundos e *Alto* acima de 2,5 segundos, conforme Figura 50.

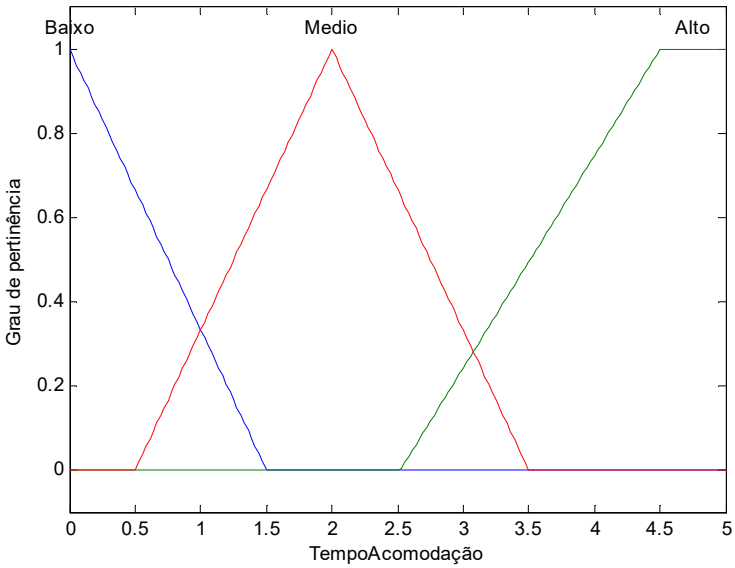


Figura 50 - Função de pertinência da variável de entrada Tempo de Acomodação.

As FP's para a saída são iguais as utilizadas nos estudos de casos anteriores.

As regras foram criadas de forma a minimizar os três objetivos. Um valor *Alto* em qualquer um dos objetivos é considerado *Ruim* e da mesma forma um valor *Muito Alto* é considerado *Muito Ruim*. As 11 regras criadas para o agregador fuzzy aplicado a sistemas de controle são mostradas na
Tabela 19.

Tabela 19 – Regras do agregador fuzzy para sistemas de controle.

Overshoot	***Tempo de Subida***	***Tempo de Acomodação***	***Aptidão***
Baixo	Baixo	-	Muito Bom
Baixo	Médio	-	Bom
Médio	Baixo	-	Bom
Médio	Médio	-	Médio
Alto	-	-	Ruim
Muito Alto	-	-	Muito Ruim
-	Alto	-	Ruim
-	Muito Alto	-	Muito Ruim
-	-	Baixo	Muito Bom
-	-	Médio	Bom
-	-	Alto	Muito Ruim

A evolução é realizada considerando o cromossomo representando os três parâmetros, *Kp*, *Ki* e *Kd*, respectivamente. O intervalo de busca utilizado é compreendido entre 0 e 100.

Os parâmetros utilizados para configuração da evolução do AG estão na Tabela 20.

Tabela 20 – Parâmetros de evolução do AG para sistemas de controle.

Parâmetro	*Valor*
Número de gerações	100
Número de indivíduos da população	100
Taxa de Cruzamento	70%
Taxa de Mutação	1%

Para as plantas analisadas foram feitas comparações com os resultados de um AG com avaliação tradicional e com resultados de técnicas tradicionais para sintonia de parâmetros.

4.5.1 Planta de 2ª ordem

Uma planta de 2ª ordem (11) foi utilizada como primeiro estudo de caso (OGATA, 2003).

$$G(s) = \frac{4}{s^2 + 0{,}5s} \tag{11}$$

A Figura 51 ilustra o sistema de controle implementado no *Simulink*, com a aplicação de um degrau unitário na entrada e é desejável que o controlador possibilite a obtenção de uma resposta próxima deste sinal de entrada aplicado.

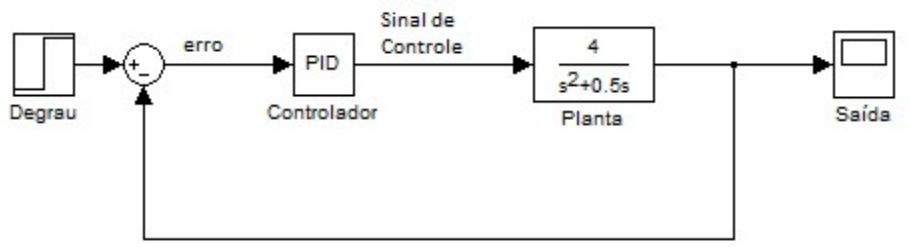


Figura 51 - Diagrama do sistema de controle de 2ª ordem.

Em OGATA (2003) é desenvolvido um compensador analítico $G_C(s)$ para este sistema cuja função de tranferência é:

$$G_C(s) = \frac{10(2s+1)(5s+1)}{(0{,}1992s+1)(80{,}19s+1)} \tag{12}$$

Neste trabalho foi executado um algoritmo genético mono objetivo com a finalidade de minimizar o RMSE e o algoritmo multiobjetivo utilizando o agregador fuzzy para minimizar o *overshoot*, o tempo de subida e o tempo de acomodação. O compensador analítico apresentado em (12) foi usado para comparação com os controladores PID obtidos pelos algoritmos genéticos.

A Tabela 21 apresenta os valores dos ganhos encontrados pelos dois AGs:

Tabela 21 – Comparação de ganhos para planta de 2ª ordem.

	AG Mono objetivo	*AG com Agregador Fuzzy – Três objetivos*
Kp	100	99,9996
Ki	0,0001	0,2436
Kd	4,7514	8,3985

A Tabela 22 apresenta os valores para o *overshoot*, o tempo de subida e o tempo de acomodação obtidos pelos três métodos analisados.

Tabela 22 – Comparação dos parâmetros de avaliação para planta de 2ª ordem.

	Compensador Analítico	*AG Mono objetivo*	*AG com Agregador Fuzzy - Três objetivos*
Overshoot (%)	21,1612	17,1186	0,4982
Tempo de subida (s)	0,3159	0,0806	0,1342
Tempo de acomodação (s)	3,4010	0,4041	0,2123

A Figura 52 apresenta a resposta obtida em função do tempo em uma simulação de 15 segundos realizada no *Simulink* com a aplicação de um degrau unitário na entrada. Os três métodos analisados são mostrados na figura.

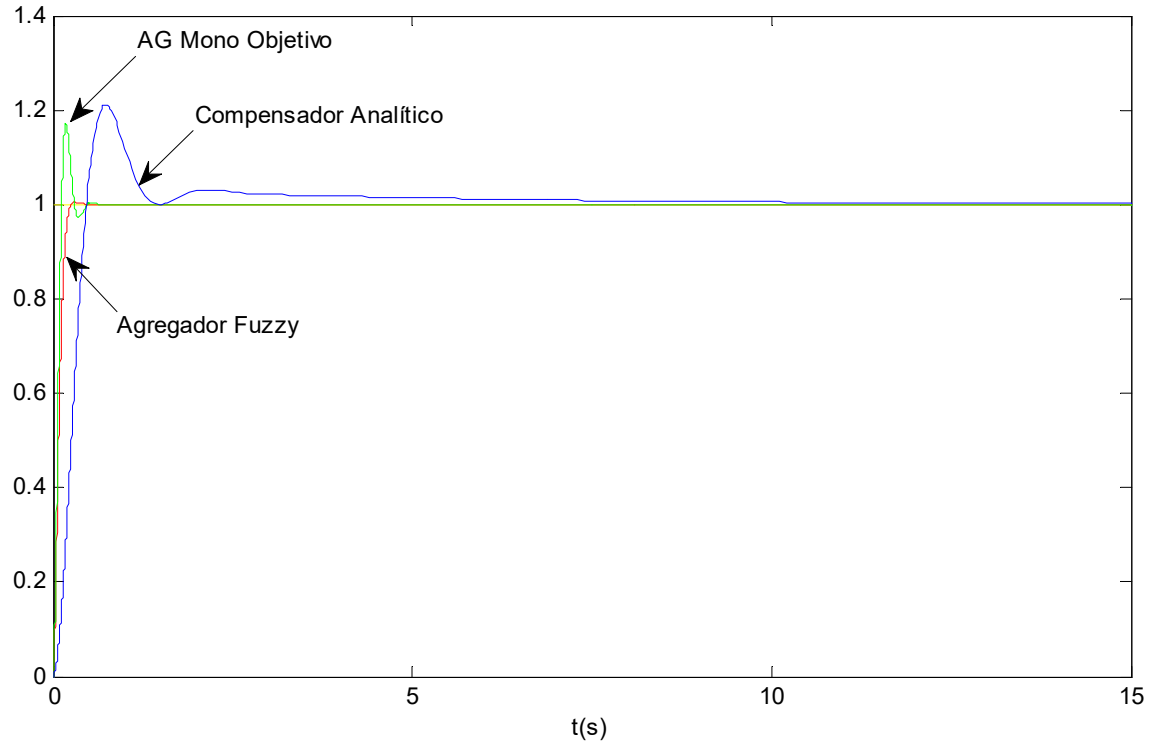


Figura 52 - Resposta a um degrau unitário pelas três técnicas analisadas.

Através da Figura 52 e dos valores apresentados na Tabela 22 observa-se que a resposta obtida pelo AG com agregação fuzzy obteve o menor *overshoot* e tempo de acomodação. O tempo de subida alcançado pelo AG mono objetivo foi o menor, porém o *overshoot* foi muito maior que o obtido pelo agregador fuzzy, o que não é desejável. O compensador analítico obteve valores mais elevados nos três parâmetros analisados.

Deste modo, os resultados obtidos mostram que o método de agregação fuzzy conseguiu minimizar os três parâmetros de forma adequada e satisfatória obtendo bons resultados em comparação aos demais controladores.

4.5.2 Planta de 3ª ordem

O segundo estudo de caso de controle corresponde a uma planta de 3ª ordem (13) apresentada em (OGATA, 2003).

$$G(s) = \frac{1}{s(s+1)(s+5)} \tag{13}$$

A Figura 53 mostra um diagrama do sistema de controle proposto, onde um degrau unitário é aplicado na entrada do sistema em malha fechada.

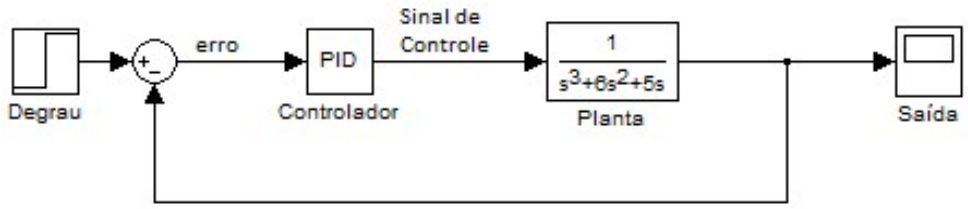


Figura 53 – Diagrama do sistema de controle de 3ª ordem.

Foram usados para comparação os valores de ganho obtidos através do método de Ziegler-Nichols e estes valores ajustados por um especialista (OGATA, 2003). Além disso, foram considerados os valores encontrados pelo algoritmo genético de um objetivo utilizando a minimização do RMSE para avaliação.

A comparação entre os valores de ganho obtidos para os quatro métodos analisados pode ser vista na Tabela 23:

Tabela 23 – Comparação de ganhos para planta de 3ª ordem.

	Ziegler-Nichols	*Ziegler-Nichols ajustado*	*AG Mono objetivo*	*Agregador Fuzzy Três objetivos*
Kp	18	39,42	49,9999	49,9995
Ki	12,81	12,81	0,0740	0,0069
Kd	6,32	30,32	20,6995	23,6521

A Tabela 24 apresenta os valores para o *overshoot*, tempo de subida e tempo de acomodação obtidos com os ganhos mostrados na Tabela 23.

Tabela 24 – Comparação dos parâmetros de avaliação para planta de 3ª ordem.

	Ziegler-Nichols	*Ziegler-Nichols ajustado*	*AG Mono objetivo*	*Agregador Fuzzy Três objetivos*
Overshoot (%)	75,4927	17,2373	10,1298	2,4089
Tempo de subida (s)	0,6523	0,8904	0,5469	0,5999
Tempo de acomodação (s)	10,4082	8,3869	2,2542	2,0993

A Figura 54 apresenta a resposta obtida em função do tempo em uma simulação realizada no *Simulink* para a aplicação de um degrau unitário na entrada com os valores de ganhos obtidos pelos quatro métodos. O tempo total de simulação é igual a 20 segundos.

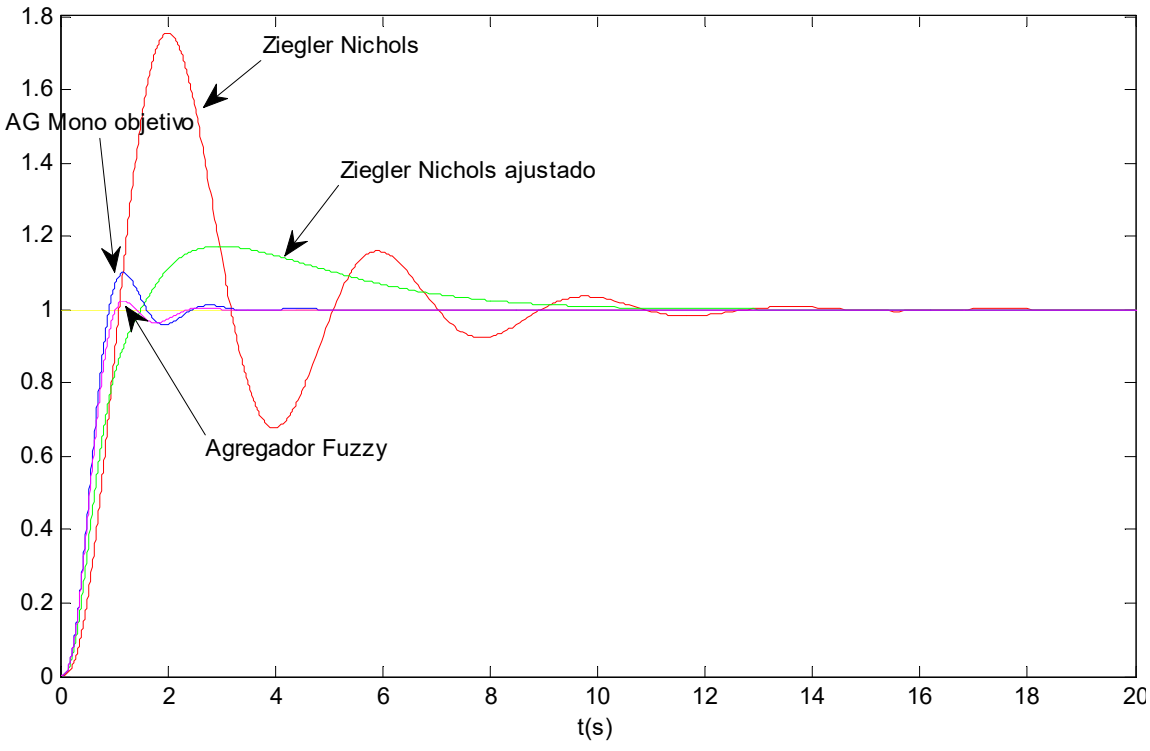


Figura 54 – Resposta a um degrau unitário pelas quatro técnicas de sintonia.

Os resultados obtidos mostram que o método de agregação fuzzy conseguiu minimizar os três parâmetros de forma adequada. Comparado aos outros métodos obteve o menor *overshoot* e tempo de acomodação. O menor tempo de subida foi obtido pelo AG mono objetivo, porém foi bem próximo do tempo obtido pelo agregador fuzzy. O método de Ziegler Nichols obteve os maiores valores para os três parâmetros analisados, porém após o ajuste conseguiu uma diminuição dos valores. Novos ajustes poderiam ser realizados, porém esta não é uma tarefa trivial e requer grande conhecimento do especialista.

Desta forma, conclui-se que o método de agregação fuzzy conseguiu obter valores bons para os ganhos de um controlador PID gerando um sistema de controle adequado.

4.5.3 Motor DC

Este estudo de caso envolve uma planta que representa um servomotor de corrente contínua controlado pela armadura com características não-lineares de saturação.

Os motores de corrente contínua (CC) são amplamente utilizados em servossistemas e são chamados de servomotores de corrente contínua.

Estes servomotores possuem a inércia do motor bastante reduzida, o que resulta em motores com alta relação entre o torque e o momento de inércia. Servomotores CC de baixa potência possuem aplicação em instrumentos e equipamentos periféricos de computadores, enquanto servomotores de média e alta potência são aplicados em sistemas robóticos, por exemplo.

A Figura 55 apresenta o modelo de um servossistema. O objetivo do sistema é controlar a posição da carga mecânica de acordo com a posição de referência.

O sistema opera com um par de potenciômetros que agem como dispositivo de erro, em que convertem as posições de entrada e saída em sinais elétricos proporcionais. O sinal de erro é amplificado pelo amplificador de ganho K_1 e a tensão é aplicada ao circuito de armadura do motor CC. Uma tensão fixa é aplicada

ao enrolamento de campo e quando existe um erro, o motor desenvolve um torque para girar a carga de forma a reduzir o erro a zero.

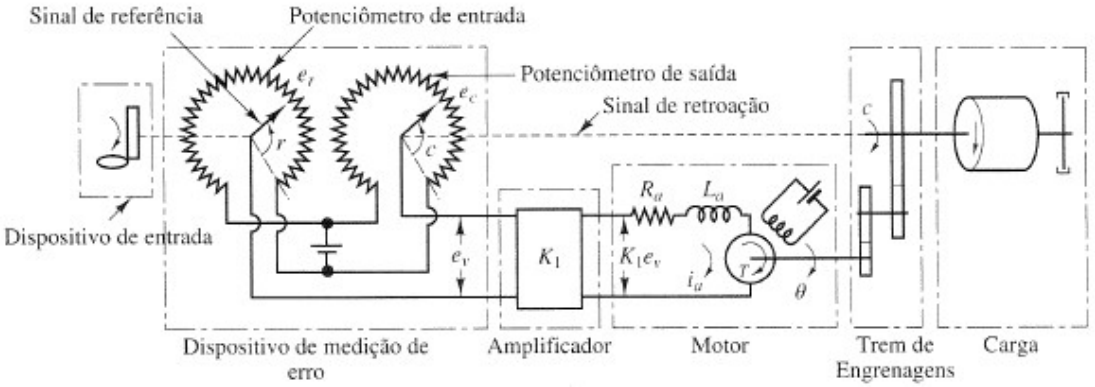


Figura 55 – Diagrama do servossistema de um motor de corrente contínua (OGATA, 2003).

Os procedimentos tradicionais para determinar as soluções de problemas envolvendo sistemas não lineares são, geralmente, bastante complexos. Para reduzir esta complexidade, muitas vezes utiliza-se um processo de linearização no qual o sistema é substituído por um sistema linear equivalente, válido para uma determinada faixa de operação.

Toda a descrição do funcionamento do motor e o desenvolvimento para obtenção da função de transferência estão descritos em detalhes em (OGATA, 2003).

Desta forma, a função de transferência do sistema é descrita pela equação (14):

$$G(s) = \frac{K_0 K_1 K_2 n}{s\left[\left(L_a s + R_a\right)\left(J_0 s + b_0\right) + K_2 K_3\right]} \tag{14}$$

Como o valor de L_a (indutância do enrolamento da armadura) é normalmente pequeno, ele pode ser desprezado. Assim, pode-se simplificar a função de transferência segundo a equação (15):

$$G(s) = \frac{K_m}{s(T_m s + 1)} \tag{15}$$

onde:

$$K_m = \frac{nK_0K_1K_2}{R_a b_0 + K_2K_3} \tag{16}$$

e

$$T_m = \frac{R_a J_0}{R_a b_0 + K_2K_3} \tag{17}$$

A definição dos parâmetros do modelo e das equações e os seus valores utilizados são conforme mostrados a seguir (OGATA, 1997):

r = deslocamento angular do eixo de entrada

c = deslocamento angular do eixo de saída

θ = deslocamento angular do eixo do motor

K_0 = ganho do potenciômetro detector de erro = $24/\pi$

K_1 = ganho do amplificador = 10

e_a = tensão elétrica aplicada ao círculo de armadura

e_b = força eletromotriz

R_a = resistência do enrolamento de armadura = $0,2\ \Omega$

L_a = indutância do enrolamento de armadura = desprezível

i_a = corrente no enrolamento de armadura

K_3 = constante de força eletromotriz = $5,5\times10^{-2} Vs/rad$

K_2 = constante de torque do motor = $6\times10^{-5} Nm/A$

J_m = momento de inércia do motor referido ao eixo do motor = $1\times10^{5} kgm^2$

b_m = coeficiente de atrito viscoso do motor referido ao eixo do motor = desprezível

J_L = momento de inércia da carga referido ao eixo de saída = $4,4\times10^{-3} kgm^2$

b_L = coeficiente de atrito viscoso da carga referido ao eixo de saída = $4\times10^{2} Nm/(rad/s)$

n = relação de engrenagens $N_1 / N_2 = 1/10$

O momento de inércia equivalente J_0 e o coeficiente de amortecimento viscoso b_0, referidos ao eixo do motor são calculados conforme equações a seguir:

$$J_0 = J_m + n^2 J_L \tag{18}$$

$$b_0 = b_m + n^2 b_L \tag{19}$$

Com estes valores, a função de transferência do motor CC é dada por:

$$G(s) = \frac{5{,}5}{s(0{,}13s + 1)} \tag{20}$$

Na Figura 56 é apresentado o diagrama de blocos do servossistema do motor CC com realimentação negativa e degrau aplicado a entrada. Este sistema foi implementado no *Simulink* e utilizado para avaliação.

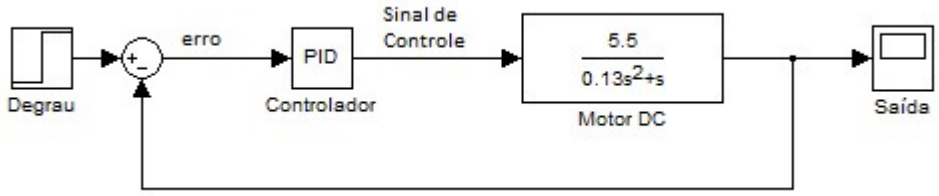


Figura 56 – Diagrama do sistema de controle para servossistema do motor CC.

Para este estudo foram obtidos quatro controladores através de técnicas mono objetivo e multiobjetivo:

- Controlador PID sintonizado por AG mono objetivo (minimização do RMSE);

- Controlador PID sintonizado por AG multiobjetivo com Agregador Fuzzy (três objetivos: minimização do *overshoot*, do tempo de subida e do tempo de acomodação);

- Controlador Fuzzy sintetizado por AG mono objetivo (minimização do RMSE);

- Controlador Fuzzy sintetizado por AG multiobjetivo com Agregador Fuzzy (três objetivos: minimização do *overshoot*, do tempo de subida e do tempo de acomodação).

Controladores PID

Os ganhos obtidos após a evolução para os dois controladores PID são mostrados na Tabela 25.

Tabela 25 – Comparação dos ganhos PID para planta do motor CC.

	Controlador PID – AG Mono objetivo	*Controlador PID – Agregador Fuzzy com três objetivos*
Kp	18,7994	19,9213
Ki	0	0,0097
Kd	0,6250	0,9747

Controladores Fuzzy

A base de regras dos sistemas fuzzy de controle foram evoluídas com os mesmos parâmetros utilizados para os controladores PID e mostrados na Tabela 20. As FP's utilizadas são mostradas na Figura 57 e na Figura 58. As duas variáveis de entrada correspondem ao erro e a variação do erro e possuem funções de pertinência iguais.

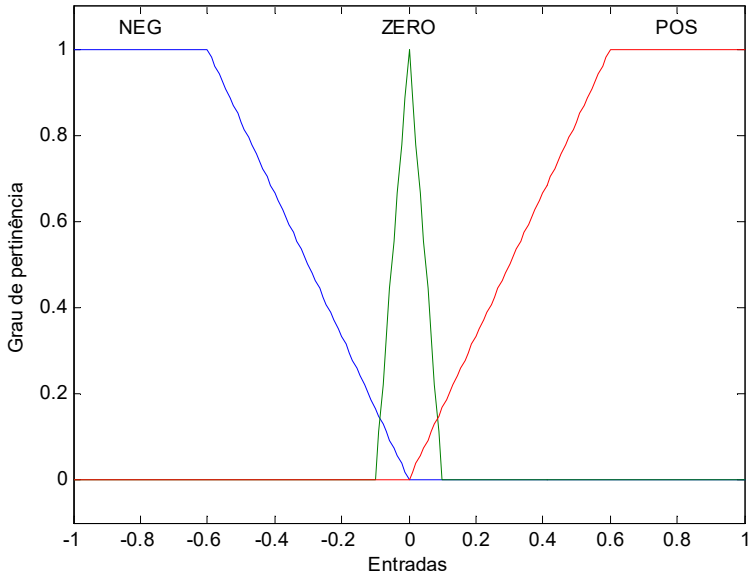


Figura 57 – Funções de pertinência das entradas (erro e variação do erro) para controle do motor CC.

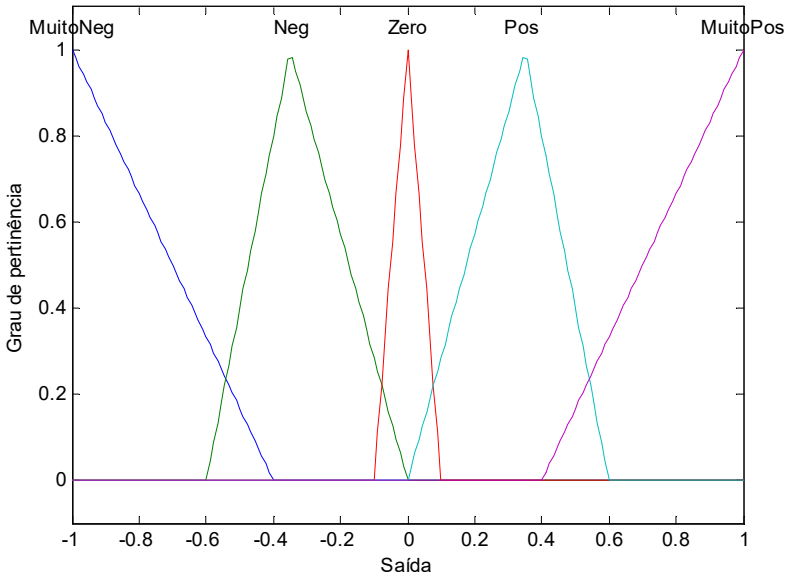


Figura 58 – Funções de pertinência da saída para controle do motor CC

As regras que foram obtidas após a evolução são mostradas na Tabela 26 para a avaliação tradicional apenas do RMSE e na Tabela 27 para a avaliação multiobjetivo com o agregador fuzzy. Uma comparação entre as duas tabelas mostra que as regras obtidas em cada modelo de avaliação são diferentes.

Tabela 26 – Regras para controlador fuzzy com avaliação do RMSE.

Variação do Erro / **Erro**	*NEG*	*ZER*	*POS*
NEG	MuitoNeg	MuitoNeg	MuitoNeg
ZER	MuitoNeg	Zero	Pos
POS	MuitoPos	MuitoPos	Zero

Tabela 27 – Regras para controlador fuzzy com avaliação do agregador fuzzy.

Variação do Erro / **Erro**	*NEG*	*ZER*	*POS*
NEG	Zero	Neg	Zero
ZER	MuitoNeg	Zero	MuitoPos
POS	MuitoPos	MuitoPos	MuitoNeg

Comparação

Para os quatro controladores obtidos foram calculados os valores do *overshoot*, tempo de subida e tempo de acomodação. Estes valores são mostrados na Tabela 28.

Tabela 28 – Comparação dos parâmetros de avaliação para planta do motor CC.

	Controlador PID - AG Mono objetivo	*Controlador PID - Agregador Fuzzy com três objetivos*	*Controlador Fuzzy - AG Mono objetivo*	*Controlador Fuzzy - Agregador Fuzzy com três objetivos*
Overshoot	8,7957%	0,4998%	3,0884%	0
Tempo de subida	0,0664s	0,0913s	0,5226	0,53333
Tempo de acomodação	0,2096s	0,1441s	0,7730	0,6847

Na Figura 59 é mostrada uma comparação entre a resposta transiente dos quatro controladores para aplicação de um degrau unitário em uma simulação de 10 segundos.

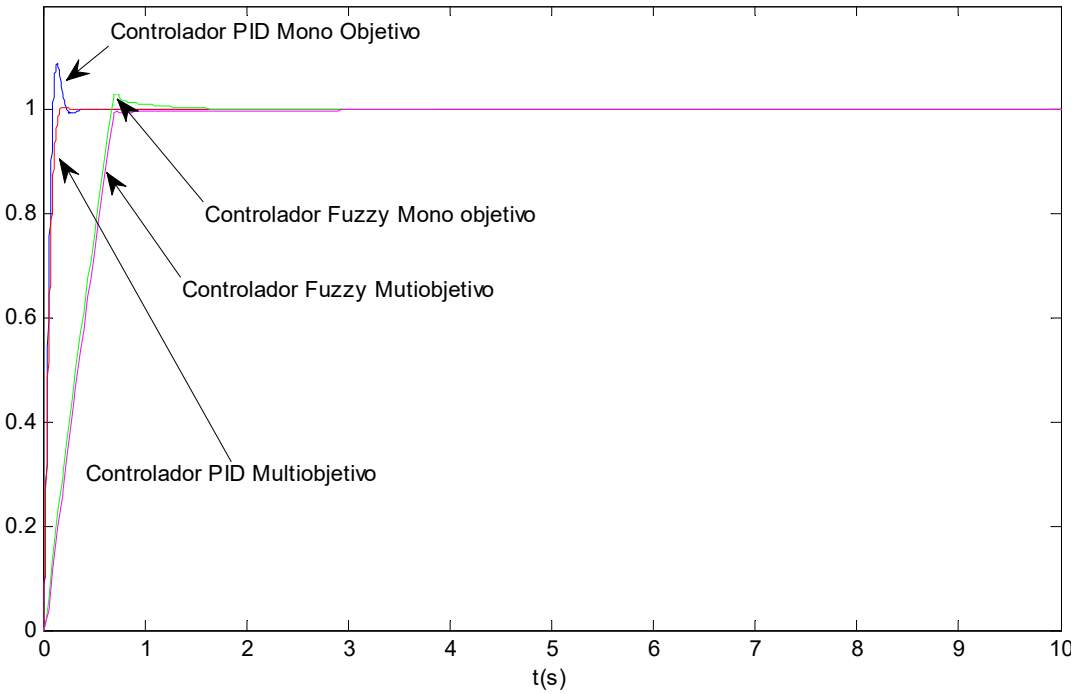


Figura 59 – Resposta ao degrau unitário obtida pelos quatro controladores.

Com os resultados obtidos é possível observar que todos os controladores obtidos atenderam aos objetivos e resultaram em controladores estáveis com erro zero em regime permanente. Os controladores PID resultaram em um tempo de resposta mais rápido que os controladores fuzzy. Por outro lado, é observado que o controlador fuzzy multiobjetivo não apresentou *overshoot*. Dependendo do modelo ao qual o motor CC será aplicado, o fato de ser um sistema mais lento pode não ser considerado um problema e a inexistência do *overshoot* pode ser um bom diferencial desejado.

Além disso, a partir de todos os estudos de casos analisados para sintonia de controladores, observa-se que quando é avaliado apenas o RMSE há uma tendência a minimizar o tempo de subida, pois ele reduz significativamente o RMSE. Portanto, quando este sistema é comparado aos sistemas obtidos pelo agregador fuzzy, apresenta um tempo de subida menor, o que não significa que o tempo encontrado pelo agregador fuzzy é ruim.

4.6 Aplicação na plataforma com hardware microcontrolado

Neste estudo de caso, foi desenvolvido um exemplo de aplicação da plataforma com hardware microcontrolado, utilizando um sistema fuzzy ajustado por um algoritmo genético aplicado a um sistema de controle.

O exemplo utilizado se refere ao controle de temperatura de um aquecedor e é baseado no módulo didático de temperatura 2302, produzido pela Datapool Eletrônica. O objetivo da planta é demonstrar didaticamente a operação dos diversos processos que envolvem o segmento da indústria no que se refere à identificação, modelagem e controle de sistemas dinâmicos de aquecimento.

O módulo vem equipado com elemento de aquecimento resistivo, amplificador (driver) *PWM* tiristorizado para acionamento do elemento de aquecimento, transdutor de temperatura a semicondutor e condicionadores de sinais (amplificadores, filtros e conversores tensão/corrente, corrente/tensão) para o transdutor e o *driver*. Além disso, o medidor de temperatura integra conector de acesso dos sinais do módulo (0 a 5 V e 0 a 20 mA) para ligação a controladores.

Em (VERLY et al., 2010) é descrito o funcionamento do módulo de temperatura e é realizado um estudo da dinâmica do processo envolvido, obtendo um modelo aproximado do sistema associado a uma lâmpada de aquecimento.

Além disso, são apresentados resultados obtidos por controladores PI ajustados por diferentes métodos. Esses resultados são utilizados neste trabalho para comparação.

A malha de controle do sistema é apresentada na Figura 60.

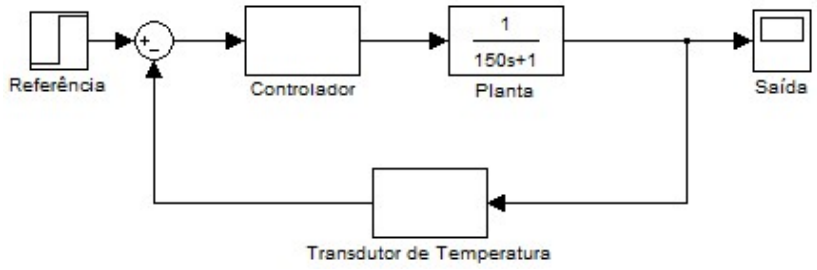


Figura 60 – Malha de controle do sistema aquecedor de temperatura.

O objetivo neste trabalho é projetar um controlador fuzzy para o sistema de aquecimento. Para isso, inicialmente, foi ajustado um controlador PI com o agregador fuzzy considerando três objetivos: a minimização do *overshoot*, do tempo de subida e do tempo de acomodação. As regras e funções de pertinência do agregador fuzzy são semelhantes às utilizadas na sintonia dos controladores PID apresentados na seção anterior.

Com isso, os ganhos obtidos foram *Kp=50* e *Ki=0,3906*.

A partir do controlador PI sintonizado, foi realizada uma simulação no *Simulink* e dados de entrada e saída do controlador foram coletados. Estes dados foram utilizados na síntese do sistema fuzzy, com o objetivo de projetar um controlador fuzzy que imite o comportamento deste controlador PI sintonizado.

Para o projeto do sistema fuzzy foi utilizado um algoritmo genético com os parâmetros mostrados na Tabela 29 e com o objetivo de minimizar o RMSE e o número de regras.

Tabela 29 – Parâmetros de configuração do AG para o controlador.

Parâmetro	*AG para Regras*	*AG para Sintonia*
Tamanho da população	100	200
Número de gerações	10	100
Taxa de crossover	50%	75%
Taxa de mutação	1%	5%

O erro do sistema e a taxa de variação do erro são utilizados como entradas do sistema fuzzy e a saída é um sinal de controle aplicado ao módulo de temperatura. O melhor sistema fuzzy obtido possui 5 funções de pertinência para a entrada “Erro”, 7 para a entrada “Variação do erro”, 7 para a saída e 33 regras. As funções de pertinência para as entradas e para a saída são mostradas na Figura 61, na Figura 62 e na Figura 63, respectivamente.

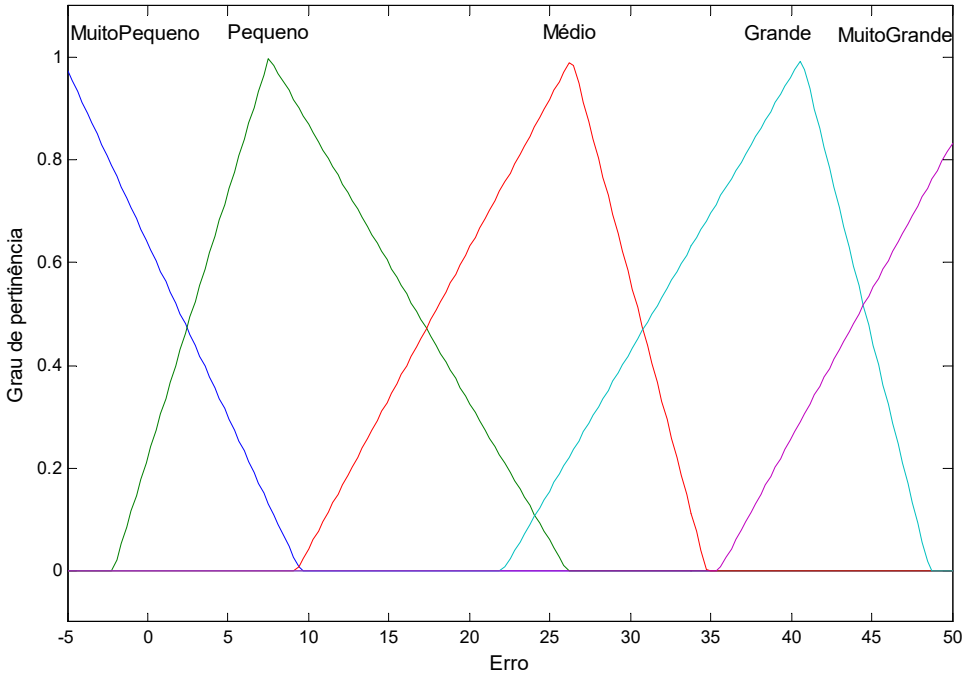


Figura 61 – Funções de pertinência para a entrada ‘Erro’.

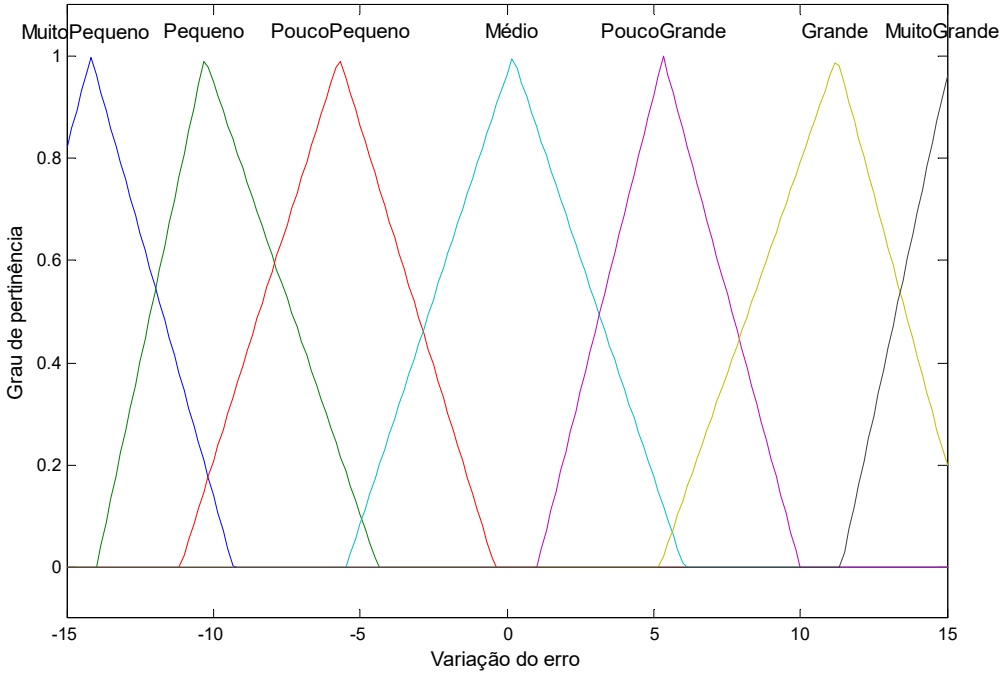


Figura 62 - Funções de pertinência para a entrada ‘Variação do erro’.

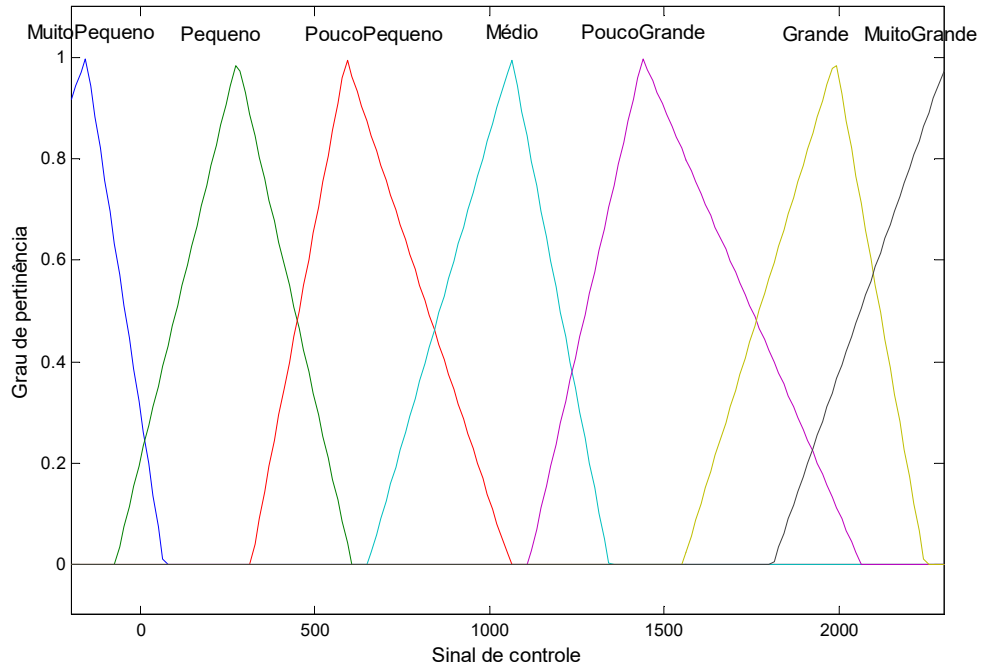


Figura 63 - Funções de pertinência para a saída ‘Sinal de Controle’.

A Tabela 30 mostra uma comparação entre os valores de *overshoot*, tempo de subida e tempo de acomodação de alguns dos métodos apresentados em (VERLY et al., 2010): o método de sintonia de Ziegler-Nichols, o método da resposta em frequência, o método do lugar das raízes, o método da sintonia de lambda, o AG com minimização do *overshoot*, o AG com minimização do tempo de acomodação e o AG com minimização do tempo de subida. Além desses, os valores obtidos com o controlador PI multiobjetivo sintonizado com uma avaliação fuzzy e o controlador fuzzy obtido através do controlador PI desenvolvidos neste trabalho também são apresentados.

Tabela 30 – Comparação entre os resultados obtidos por diferentes métodos.

Método		***Overshoot (%)***	***Tempo de Subida (s)***	***Tempo de Acomodação (s)***
Controladores PI	***Método de Ziegler-Nichols***	5,31	160,90	527,42
	Método da resposta em	11,04	9,50	64,49
	Método do Lugar das raízes	**0**	64,54	115,18
	Sintonia de lambda	**0**	10,99	45,00
	AG overshoot	**0**	65,35	545,67
	AG tempo de subida	20,28	28,63	180,70
	AG tempo de acomodação	**0**	8,40	15,87
	AG multiobjetivo	0,28	6,52	11,33
Controlador Fuzzy		0,03	**6,31**	**11,29**

Os resultados obtidos pela avaliação multiobjetivo foram bons, pois valores menores que os obtidos pelos métodos tradicionais foram alcançados em dois dos objetivos analisados. Além disso, o *overshoot* apresenta valor muito baixo, próximo do mínimo obtido por outros métodos. O controlador fuzzy conseguiu reproduzir o comportamento do controlador PI ajustado pelo AG multiobjetivo e ainda obteve valores um pouco menores para os três parâmetros.

Assim, com o controlador fuzzy projetado, foi iniciada a implementação do hardware para teste do sistema. No hardware implementado não foi utilizado o módulo de temperatura, pois devido à falta de recursos não foi possível a sua aquisição. Além disso, como o objetivo neste trabalho é implementar uma plataforma que possibilite a aquisição e envio de dados e testar o seu comportamento junto à um sistema inteligente, é possível testar de uma forma mais dinâmica a aquisição de

dados e o comportamento do sistema, através de potenciômetros para as entradas e da medição da tensão de saída com um multímetro.

Desta forma, foram utilizados dois potenciômetros de 10 $k\Omega$ nas entradas, conectados ao microcontrolador Arduino. Cada potenciômetro representa uma das entradas do sistema fuzzy. Um dos potenciômetros simula o erro da temperatura do ambiente em relação ao *setpoint* desejado e o outro a variação do erro. Na saída foi usado um filtro passa baixa composto por um resistor de 4,7 $k\Omega$ e um capacitor de 1 μ F a fim de filtrar a saída PWM do Arduino. Os valores do resistor e capacitor foram calculados de forma que a frequência do filtro seja muito menor do que a frequência PWM do Arduino que oscila em aproximadamente 500 kHz. Dessa forma, o filtro utilizado possui uma frequência de corte de aproximadamente 30 kHz, que para o teste mostrou-se adequada. O circuito implementado e as conexões estão ilustradas na Figura 64.

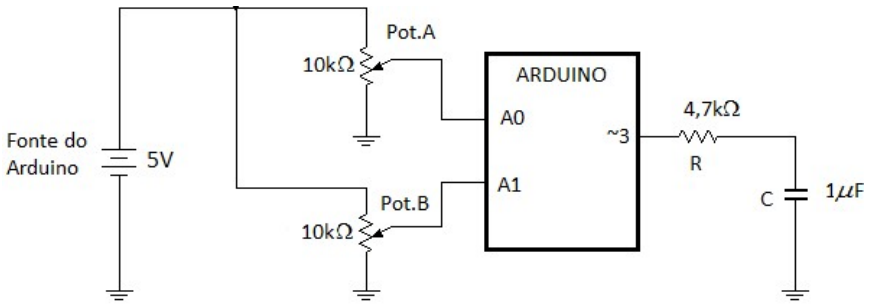


Figura 64 – Circuito implementado.

No Simulink, foi desenvolvida uma interface para interligação do controlador fuzzy, que é executado no Matlab, com o microcontrolador Arduino. O ambiente criado no Simulink é mostrado na Figura 65.

Os blocos de ganho *1* e *2* são adicionados para modificar os valores das entradas que são obtidas pelo Arduino na escala de 0 a 1023 para valores entre 0 e 5 Volts com a finalidade de possibilitar conferência com os valores medidos de uma forma mais simples. Para os dois blocos foram utilizados o valor de ganho igual a 4,92/1023. O numerador desta constante foi determinado através de medições realizadas no Arduino, pois foi verificado que a tensão fornecida era igual a 4,92 V.

Os blocos *ganho 3* e *ganho 4*, as *constantes 1* e *2* e os somadores são necessários para modificar os valores para escalas correspondentes aos limites

especificados. Tal modificação tem o intuito de facilitar a visualização e compatibilizar com os limites especificados para as variáveis de entrada do sistema fuzzy.

Da mesma maneira, o bloco *ganho 5* e a *constante 3* são responsáveis por mudar a escala, transformando o sinal de controle em valores de 0 a 5 V. O valor do ganho é igual a 4,92/2500. O bloco *ganho 6* modifica os valores de 0 a 5 para uma escala de 0 a 255 correspondentes à saída PWM. Para este bloco foi utilizado um ganho igual a 255/4,92.

Os blocos *ganho 1* e *ganho 3*, *ganho 2* e *ganho 4* e *ganho 5* e *ganho 6* poderiam ser unificados em blocos únicos, porém foi escolhido manter dois blocos separados por motivos didáticos.

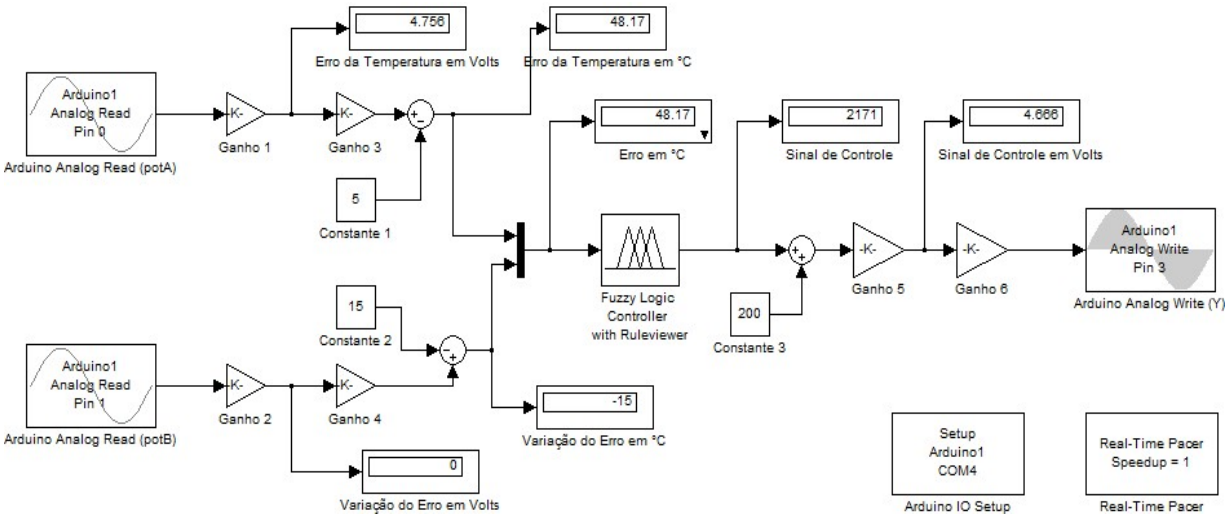


Figura 65 – Implementação no Simulink.

A partir deste modelo foram realizados testes para comprovação da funcionalidade do sistema.

O potenciômetro A representa a entrada “Erro”. O potenciômetro foi ajustado em diferentes posições, simulando diferentes valores de erro em relação a temperatura desejada.

O potenciômetro B representa a entrada “Variação do erro” e também pode ser alterado para simulação de diferentes situações.

A Tabela 31 apresenta 10 valores de tensão aplicados nas entradas através da variação dos dois potenciômetros. Junto aos valores de tensão são apresentados

os correspondentes erros de temperatura em °C ou a variação do erro, que representam os valores utilizados nas entradas do sistema fuzzy nas escalas correspondentes. Os valores de tensão esperados para a saída do sistema fuzzy, os valores de tensão medidos na saída e os erros equivalentes também são mostrados na tabela.

Tabela 31 – Resultados para valores de teste.

Entradas		***Saída***		
Tensão no potenciômetro A em V (Erro em °C)	*Tensão no potenciômetro B em V (Variação do erro)*	*Tensão Esperada (V)*	*Tensão Obtida (V)*	*Erro (%)*
4,473 (45)	0,149 (-14,1)	4,615	4,61	0,108
4,04 (40,16)	0,313 (-13,09)	4,253	4,25	0,070
3,299 (31,88)	0,818 (-10,01)	3,410	3,40	0,293
2,688 (25,05)	1,01 (-8,84)	2,987	2,98	0,234
2,284 (20,54)	1,414 (-6,38)	2,282	2,28	0,088
1,707 (14,09)	1,525 (-5,70)	2,101	2,10	0,047
1,284 (9,35)	1,828 (-3,86)	1,467	1,47	-0,204
0,971 (5,86)	2,193 (-1,63)	1,013	1,01	0,296
0,707 (2,90)	2,284 (-1,07)	0,788	0,79	-0,254
0,491 (0,48)	2,443 (-0,103)	0,532	0,53	0,301

Através dos dados obtidos conclui-se que a plataforma e o sistema de controle implementado apresentaram o comportamento esperado. Para os valores analisados foram calculados os erros relativos porcentuais entre os valores de tensão esperados e os valores de tensão medidos. O maior erro obtido é igual a 0,301%, o que é aceitável e pode ser explicado pela precisão do multímetro utilizado.

Desta forma, foi verificado que a plataforma tem potencialidade para aplicação no desenvolvimento de um controlador fuzzy real através da conexão do microcomputador com o microcontrolador Arduino e a partir deste estudo podem ser desenvolvidas novas aplicações.

5 CONSIDERAÇÕES FINAIS

5.1 Conclusões

Neste trabalho foi concebido um modelo evolucionário para o desenvolvimento de sistemas de inferência fuzzy, o qual usa um método para avaliação que considera mais de um objetivo e utiliza, para isso, um processo de agregação de objetivos através de um sistema fuzzy. Tal método foi denominado agregador fuzzy.

O agregador fuzzy foi aplicado no processo de avaliação de algoritmos genéticos, modificando o método tradicional destes algoritmos e incluindo, dessa forma, a característica de avaliação multiobjetivo a tais algoritmos evolutivos.

A fim de analisar a eficácia da técnica desenvolvida, foram realizados estudos de casos envolvendo diferentes áreas de aplicação.

O primeiro estudo de caso envolveu o projeto de roteamento em redes de sensores sem fio e teve foco no agregador fuzzy. Para isso, o agregador fuzzy foi aplicado no processo de determinação do melhor caminho para transmissão de dados considerando dois objetivos: a menor distância percorrida e a economia de energia dos nós sensores/roteadores.

O modelo desenvolvido utilizando o algoritmo genético com avaliação fuzzy foi inicialmente aplicada em problemas de minimização e maximização de funções. Foram realizados estudos relativos a maximização da *função f6* e da *função f6 deslocada* e estudos com a minimização de funções de *benchmark* multiobjetivo. Para todos os casos analisados os resultados obtidos foram comparados a resultados de outras técnicas existentes na literatura.

O projeto de sistemas fuzzy foi abordado através de dois estudos de casos utilizando bancos de dados existentes na literatura. O algoritmo genético foi utilizado para evoluir a base de regras e sintonizar as funções de pertinência, sendo que se considerou a minimização do RMSE e a minimização do número de regras, objetivos de entrada do agregador fuzzy, com a finalidade de obter sistemas eficientes e interpretáveis.

Foram realizados estudos com controladores PID e um estudo com controladores fuzzy. Os parâmetros dos controladores PID foram evoluídos através do algoritmo genético, utilizando um agregador fuzzy para avaliar a minimização de três objetivos: o *overshoot*, o tempo de subida e o tempo de acomodação. O controlador fuzzy teve a base de regras projetada e funções sintonizadas através do algoritmo genético, que considerou também os três objetivos. Foram apresentadas tabelas e gráficos comparando os controladores com avaliação multiobjetivo com aqueles com avaliação de apenas um objetivo.

Além disso, foi implementada uma interface microcontrolada simples que possibilita realizar entrada e saída com um processo ou planta real. Uma interface de conexão entre o microcomputador e um microcontrolador Arduino foi criada, permitindo aquisição e envio de dados. Para analisar o comportamento da plataforma foi evoluído um sistema fuzzy que executa um controle de temperatura. Tendo em vista que a planta real não estava disponível, potenciômetros foram utilizados para representar os valores das entradas do sistema e a tensão de saída foi medida por um multímetro. Foram feitas medições com alterações dos valores das entradas e observado o comportamento real e simulado. O protótipo da plataforma de execução em hardware apresentou o comportamento esperado e se mostrou adequado para implementações reais de sistemas inteligentes simples.

Todos os estudos de casos analisados apresentaram resultados consistentes e satisfatórios, atendendo aos objetivos e especificações previamente estabelecidos. Em comparação aos outros métodos analisados, obtiveram resultados compatíveis, com as vantagens de inserir preferências e especificações do projetista de uma forma simples e interpretável no início do projeto e não necessitar de interferência do projetista, nem durante e nem após o processo de evolução.

Dessa forma, o trabalho desenvolvido, incluindo a ferramenta de avaliação multiobjetivo através de um sistema fuzzy e a implementação da interface para aquisição de dados de uma planta real, constitui uma contribuição aos estudos de projeto e implementação de sistemas de inferência fuzzy que podem ser utilizados em diversas aplicações.

5.2 Trabalhos futuros

Para trabalhos futuros podem ser sugeridos estudos em diferentes linhas de atuação.

Uma sugestão é a realização de um estudo que viabilize alterar a atual estrutura do algoritmo desenvolvido para uma estrutura que permita a sua execução em paralelo. Esse paralelismo computacional permitiria que o processo de validação e simulação fosse acelerado, possibilitando que sistemas mais complexos sejam avaliados mais facilmente e que populações com um maior número de indivíduos sejam utilizadas.

Outra possibilidade é relacionada à aplicação de algoritmos genéticos com avaliação multiobjetivo na evolução de circuitos. Para isso, é necessário um estudo relativo à utilização de um software de simulação de circuitos em conjunto com o Matlab, possibilitando a evolução e avaliação de circuitos eletrônicos com múltiplos objetivos.

Ainda é sugerido implementar o sistema fuzzy de controle de temperatura projetado na seção 4.6, incluindo a obtenção de dados de um sensor de temperatura, como o LM35, e utilizar o módulo aquecedor como atuador.

Além disso, seria interessante selecionar um hardware microcontrolado mais poderoso, de modo a implementar estudos de casos mais complexos na plataforma, assim como viabilizar a execução do próprio algoritmo no sistema microcontrolado. Uma possibilidade seria estudar e utilizar microcontroladores mais profissionais. Por exemplo, um hardware processador baseado em CPU de baixo custo e alto desempenho, como o *Raspberry Pi*, para possibilitar um hardware com execução independente.

REFERÊNCIAS

ACHARYA, S.; SAHA, S.; THADISINA, Y. Multiobjective Simulated Annealing-Based Clustering of Tissue Samples for Cancer Diagnosis, IEEE Journal of Biomedical and Health Informatics (Volume: 20, Issue: 2, March 2016), p.691-698, 2015.

ALCALÁ, R.; DUCANGE, P.; HERRERA, F.; LAZZERINI, B. Multiobjective Evolutionary Approach to Concurrently Learn Rule and Data Bases of Linguistic Fuzzy-Rule-Based Systems, IEEE Transactions on Fuzzy Systems, Vol.17, No.5, p.1106-1122, 2009.

ALCALÁ-FDEZ, J.; FERNANDEZ, A.; LUENGO, J.; DERRAC, J.; GARCÍA, S.; SÁNCHEZ, L.; HERRERA, F. KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. Journal of Multiple-Valued Logic and Soft Computing 17:2-3, p.255-287, 2011.

ALTINOZ, O. T.; DEB, K. Late Parallelization and Feedback Approaches for Distributed Computation of Evolutionary Multiobjective Optimization Algorithms, Second International Conference on Soft Computing and Machine Intelligence, 2015.

AMARAL, J. F. Síntese de Sistemas Fuzzy por Computação Evolucionária. Tese de Doutorado, Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, 2003.

ANDERSSON, J. A survey of multiobjective optimization in engineering design. Tech. Rep. LiTH-IKP-R-1097, Department of Mechanical Engineering, Linköping University, 2000.

ARDUINO. Disponível em: <http://arduino.cc/.>. Acesso em: 2 jan. 2017.

ARROYO, J. E. C. Heurísticas e metaheurísticas para otimização combinatória multiobjetivo. Tese de doutorado, Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, 2002.

AZZABI, L; AYADI, D.; KOBI, A.; BACHAR, K.; ROBLEDO, C. The multiobjective optimization problem with fuzzy goal programming, Proceedings - Annual Reliability and Maintainability Symposium (RAMS), 2013.

BENAYOUN, R.; DE MONTGOLFIER, J.; TERGNY, J.; LARITCHEV, O. Linear programming with multiple objective functions: Step Method (STEM), Mathematical Programming, vol. 1, pp. 366-375, 1971.

BENEDICT, S. Application of Energy Reduction Techniques using Niched Pareto GA of Energy Analzyer for HPC Applications, 7th International Conference on Contemporary Computing (IC3), 2014.

BUAYAI, K.; LEKDEE, C.; INRAWONG, P.; KERDCHUEN, K. Practical Capacitor Placement for Power Loss Reduction and Voltage Improving by NSGA-II Electrical Engineering/Electronics, 13th International Conference on Computer, Telecommunications and Information Technology (ECTI-CON), 2016.

CHAMANI, M. R.; POURSHAHABI, S.; SHEIKHOLESLAM, F. Fuzzy genetic algorithm approach for optimization of surge tanks, Scientia Iranica A, Vol. 20, pp.278-285, 2013.

CHARNES, A.; COOPER, W. W. Management models and industrial applications of linear programing. New York, John Wiley & Sons, 1961.

CHENG, J.; YEN, G. G.; ZHANG, G. A grid-based adaptive multi-objective differential evolution algorithm, Information Sciences, Volumes 367–368, pp. 890–908, 2016.

CHUNG, H. S. Multidisciplinary Design Optimization of Supersonic Business Jets using Approximation Model-Based Genetic Algorithms. PhD thesis, Department of Aeronautics and Astronautics, Stanford University, California, USA, 2004.

CINTRA, M. E.; MONARD, M. C.; CAMARGO, H. A. FCA-based rule generator, a framework for the genetic generation of fuzzy classification systems using formal concept analysis, 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2015.

COELHO, P. H. G.; AMARAL, J. F. M.; ALMEIDA, N. N.; FONSECA, A. C. X. S.; GUIMARÃES, K. P. Application of Fuzzy Inference Systems in the Transmission of Wireless Sensor Networks. 19º ICEIS – International Conference on Enterprise Information Systems, 2017.

COELLO COELLO, C. A. Multi-objective evolutionary algorithms in real-world applications: some recent results and current challenges. Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences, Vol. 36 of the series Computational Methods in Applied Sciences pp 3-18, 2013.

COELLO COELLO, C. A. A comprehensive survey of evolutionary-based multiobjective optimization techniques. Knowledge and Information Systems, Volume 1, Issue 3, p. 269–308, 1999.

CORDÓN, O.; HERRERA, F.; SÁNCHEZ, L. Solving Electrical Distribution Problems Using Hybrid Evolutionary Data Analysis Techniques, Applied Intelligence 10, pp.5-24, 1999.

DAVIS, L. D. Handbook of genetic algorithms, VNR Comp. Library, 1990.

DEB, K.; JAIN, H. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems

With Box Constraints, IEEE Transactions on Evolutionary Computation, Vol. 18, Issue 4, pp. 577 – 601, 2014.
DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6(2) p.182-197. 2002.

DEB, K.; PRATAP, A.; MEYARIVAN, T. Constrained Test Problems for Multi-objective Evolutionary Optimization. In: Zitzler E., Thiele L., Deb K., Coello Coello C.A., Corne D. (eds) Evolutionary Multi-Criterion Optimization. EMO 2001.

ELSERSY, M.; AHMED, M. H.; ABDERRAZAK, A.; ELFOULY, T. M. Routing and Flow Rate Assignment Using Multi-Objective Optimization in Wireless Sensor Networks. Wireless Communications and Networking Conference (WCNC): - Track 3: Mobile and Wireless Networks, 2015.

EREMIA, M.; LIU, CHEN-CHING; EDRIS, ABDEL-ATY. Artificial Intelligence and Computational Intelligence: A Challenge for Power System Engineers, 1072p, Wiley-IEEE Press, 2016.

FARINA, M.; AMATO, P. A fuzzy definition of ‘optimality’ for manycriteria optimization problems. IEEE Trans. Syst., Man, Cybern., A Syst. Hum., vol. 34, no. 3, pp. 315–326, 2004.

FERNANDES, J. P. T. Abordagem Lexicográfica na Otimização da Operação de Usinas Hidrelétricas, Tese de Doutorado, Universidade Estadual de Campinas, 2015.

FIGUEIREDO, K. T.; VELLASCO, M. M. B. R.; PACHECO, M. A. C. ; SOUZA, F. J. DE . Hierarchical Neuro-Fuzzy Models Based on Reinforcement Learning for Autonomous Agents. International Journal of Innovative Computing, Information & Control, v. 10, p. 1471-1494, 2014.

FONSECA, A. C. X. S. Módulo microcontrolado para aplicações em sistemas fuzzy. Projeto de graduação, Universidade do Estado do Rio de Janeiro, 2014.

FONSECA, C. M.; FLEMING, P. J. Multiobjective Genetic Algorithms Made Easy: Selection, Sharing and Mating Restriction, First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, (Conf. Publ. No. 414), 1995.

FONSECA, C. M.; FLEMING, P. J., Multiobjective optimization and multiple constraint handling with evolutionary algorithms - Part I: a unified formulation, IEEE Transactions on Systems, Man, & Cybernetics Part A: Systems & Humans, vol. 28, pp. 26-37, 1998a.

FONSECA, C. M.; FLEMING, P. J., Multiobjective optimization and multiple constraint handling with evolutionary algorithms - Part II: application example, IEEE Transactions on Systems, Man, & Cybernetics Part A: Systems & Humans, vol. 28, pp. 38-47, 1998b.

GACTO, M. J.; ALCALÁ, R.; HERRERA, F. Integration of an Index to Preserve the Semantic Interpretability in the Multiobjective Evolutionary Rule Selection and Tuning of Linguistic Fuzzy Systems. IEEE Transactions on Fuzzy Systems, Vol. 18, No. 3, p. 515-531, 2010.

GACTO, M. J.; ALCALÁ, R.; HERRERA, F. Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures, Information Sciences, Special Issue on Interpretable Fuzzy Systems, Vol. 181, Issue 20, p.4340–4360, 2011.

GACTO, M. J.; GALENDE, M.; ALCALÁ, R.; HERRERA, F. Obtaining Accurate TSK Fuzzy Rule-Based Systems by Multi-Objective Evolutionary Learning in High-Dimensional Regression Problems, IEEE International Conference on Fuzzy Systems (FUZZ), 2013.

GARCÍA-MARTÍNEZ, S.; ESPINOSA-JUÁREZ, E.; RICO-MELGOZA, J. J. Expansion of Electrical Networks Considering Power Quality Aspects by Applying a Multi-objective Tabu Search Technique. International Conference on Computational Science and Computational Intelligence, 2015.

GOLDBERG, D. E. Genetic algorithms in search, optimization, and machine learning. Reading, Addison-Wesley Professional, 1989.

GONÇALVES, L. B.; PACHECO, M. A. C. Estudo comparativo dos métodos de aptidão para problemas com múltiplos objetivos. Revista ICA, nº2. 2009.

HAIMES, Y. Y.; LASDON, L. S.; WISMER, D. A. On a bicriterion formulation of the problems of integrated system identification and system optimization. IEEE Transactions on Systems, Man, and Cybernetics, SMC-1:296-297, pp.23, 32, 1971.

HAJIMANI, E.; RUANO, M. G.; RUANO, A. E. MOGA Design for Neural Networks Based System for Automatic Diagnosis of Cerebral Vascular Accidents. 9th International Symposium on Intelligent Signal Processing (WISP), 2015.

HE, Z.; YEN, G. G.; ZHANG, J. Fuzzy-Based Pareto Optimality for Many-Objective Evolutionary Algorithms, IEEE Transactions on Evolutionary Computation, Vol. 18, No. 2, pp. 269 – 285, 2014.

HERRERA, F. Genetic fuzzy systems: taxonomy, current research trends and prospects. Evolutionary Intelligence, journal, v.1, n.1, p.27-46, 2008.

HORN, J.; NAFPLIOTIS, N. Multiobjective Optimization Using the Niched Pareto Genetic Algorithm, Technical Report 93005, Illinois Genetic Algorithm Laboratory, Dept. of General Engineering, University of Illinois at Urbana- Champaign, Urbana, USA, 1993.

HOUCK, C.R.; JOINES, J.; KAY, M. A genetic algorithm for function optimization: A Matlab implementation, ACM Transactions on Mathmatical Software, 1996.

HUBAND, S.; HINGSTON, P.; BARONE, L.; WHILE, L. A Review of multiobjective Test Problems and a Scalable Test Problem Toolkit, IEEE Transactions on Evolutionary Computation, Vol. 10, No. 5, 2006.

ISHIBASHI, R. Extração de conhecimentos com interpretabilidade aumentada utilizando modelagem fuzzy e otimização multi-objetivo. Tese de Doutorado, Instituto Tecnológico da Aeronáutica, 2013.

ISHIBUCHI, H. Multiobjective genetic fuzzy systems: review and future research directions, Proceedings of the 2007 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2007), p.913-918, 2007.

JAIMES, A. L.; COELLO COELLO, C. A. Many-objective Problems: Challenges and Methods, Chapter Springer Handbook of Computational Intelligence, pp.1033-1046, 2012.

JARIYATANTIWAIT, C. Multiobjective differential evolution based on fuzzy performance feedback: soft constraint handling and its application in antenna designs, Tese de doutorado, Oklahoma State University, 2015.

JIANG, S.; YANG, S. Evolutionary Dynamic Multiobjective Optimization: Benchmarks and Algorithm Comparisons, IEEE Transactions on Cybernetics, Volume: PP, Issue: 99, pp. 1-14, 2016.

JONATHAN, M.; PACHECO, M. A. C. Técnicas de otimização de problemas com múltiplos objetivos - Um estudo sobre o método de minimização de energia e suas variantes. Departamento de Engenharia Elétrica da Pontifícia Universidade Católica do Rio de Janeiro, Revista Rica, 2010.

KELL. Disponível em: <http://www.keel.es>. Acesso em: 05 dez. 2016.

KHAJWANIYA, K. K.; TIWARI, V. Satellite image denoising using Weiner filter with SPEA2 algorithm, IEEE 9th International Conference on Intelligent Systems and Control (ISCO), 2015.

KNOWLES, J.; CORNE, D. The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Pareto Multiobjective Optimisation, Congress on Evolutionary Computation, 1999.

KÖPPEN, M.; GARCIA, R. V.; NICKOLAY, B. Fuzzy-Pareto-dominance and its application in evolutionary multiobjective optimization, Proc. Int. Conf. Evol. Multi-Criterion Optimization, pp. 399–412, 2005.

KOSHIYAMA, A. S. GPFIS: Um sistema fuzzy-genético genérico baseado em programação genética. Dissertação de mestrado, PUC-Rio, 2014.

LABATI, R. D.; GENOVESE, A.; MUÑOZ, E.; PIURI, V.; SCOTTI, F.; SFORZA, G. Computational intelligence for industrial and environmental applications, IEEE 8th International Conference on Intelligent Systems (IS), 2016.

LACERDA, E. G. M.; CARVALHO, A. C. P. L. Introdução aos algoritmos genéticos. In: Galvão, C. O., Valença, M. J. S. (orgs.) Sistemas inteligentes: aplicações a recursos hídricos e ciências ambientais. Porto Alegre: Associação Brasileira de Recursos Hídricos. p. 99-150. 1999.

LARA, A.; SANCHEZ, G.; COELLO COELLO, C. A.; SCHUTZE, O. HCS: A New Local Search Strategy for Memetic Multi-Objective Evolutionary Algorithms. IEEE Transactions on Evolutionary Computation, 14(1):112–132, 2010.

LEITE, L. da C. M., Geração e simplificação da base de conhecimento de um sistema híbrido fuzzy-genético, Dissertação de Mestrado, UERJ, 2009.

LI, X.; ZENG, S.; QIN, S.; LIU, K. Constrained Optimization Problem Solved by Dynamic Constrained NSGA-III Multiobjective Optimizational Techniques. Congress on Evolutionary Computation (CEC), 2015.

LIMA, I. R. Sistemas inteligentes para auxiliar na tomada de decisões em mercado de capitais, Dissertação de Mestrado, Universidade Federal de Lavras, 2012.

LOPEZ, E. M. Un algoritmo evolutivo multiobjetivo basado en hipervolumen en GPUs, tese mestrado, Centro de Investigacion y de Estudios Avanzados del Instituto Politécnico Nacional, 2014.

LUCA, M.; LUCA, R.; BEJINARIU, SILVIU-IOAN; CIOBANU, A.; PADURARU, O.; ZBANCIOC, M.; BARBU, T. An Overview of Several Researches on Fuzzy Logic in Intelligent Systems. 2015 International Symposium on Signals, Circuits and Systems (ISSCS), 2015.

LUONG, T. V.; MELAB, N.; TALBI, E-G. GPU-Based Approaches for Multiobjective Local Search Algorithms. A Case Study: The Flowshop Scheduling Problem. In Evolutionary Computation in Combinatorial Optimization, 11th European Conference, pp.155–166. Lecture Notes in Computer Science Vol. 6622, 2011.

MAMDANI, E. H.; ASSILIAN, S. An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. International Journal of Man-Machine Studies, 7(1): 1-13, 1975.

MARTÍNEZ, S. Z.; COELLO COELLO, C. A. A Hybridization of MOEA/D with the Nonlinear Simplex Search Algorithm. In Proceedings of the 2013 IEEE Symposium on Computational Intelligence in Multicriteria Decision Making (MCDM'2013), pp.48–55, Singapore, 2013.

MATEUS, G. R.; RESENDE, M. G. C.; SILVA, R. M. A. GRASP: Procedimentos de busca gulosos, aleatórios e adaptativos. 2ª Escola Luso-Brasileira de Computação Evolutiva, Portugal, 2010.

MATLAB – The Language of Technical Computing – MathWorks. Disponível em: <http://www.mathworks.com>. Acesso em: 20 dez. 2016.

MENG, Q.; QIAO, J.; YANG, C. Multi-objective design of the Water distribution systems using SPEA2, 35th Chinese Control Conference (CCC), 2016.

MUKHERJEE, J.; CHOWDHURI, S. Multiobjective optimization by PSO for Switched Reluctance Motor (SRM) drive, 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO), 2014.

NASIR, M.; MONDAL, A. K.; SENGUPTA, S.; DAS, S.; ABRAHAM, A. An improved multiobjective evolutionary algorithm based on decomposition with fuzzy dominance, Proc. IEEE Congr. Evol. Comput., USA, pp. 765–772, 2011.

NEOCLEOUS, A. C.; NICOLAIDES, K. H.; SCHIZAS, C. N. First Trimester Noninvasive Prenatal Diagnosis: A Computational Intelligence Approach, IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, vol. 20, No. 5, 2016.

OGATA, K. Engenharia de Controle Moderno. 4ª ed., Prentice Hall do Brasil, 2003.

OGATA, K. Engenharia de Controle Moderno. 3ª ed. Prentice Hall do Brasil, 1997.

OLIVEIRA, L. S. Uma Contribuição ao Estudo dos Métodos de Otimização Multi-Objetivo. Dissertação de mestrado, Universidade Federal de Uberlândia, 2005.

OMAR, M. T.; GOPE, M.; KHANDAKER , A. I.; SHILL, P. C. Multi Objective Non-dominated Sorting Genetic Algorithm (NSGA-II) for Optimizing Fuzzy Rule Base System. Proceedings of International Conference on Electrical Information and Communication Technology (EICT), 2015.

OSMAN, M.S.; MAHMOUD A. ABO-SINNA; MOUSA, A.A. A combined genetic algorithm-fuzzy logic controller (GA–FLC) in nonlinear programming, Applied Mathematics and Computation, Vol. 170 Issue 2, pp.821-840, 2005.

OSYCZKA, A. Multicriteria optimization for engineering design. John S. Gero, editor, Design Optimization, p. 193-227. Academic Press, 1985.

PAL, B. B.; SEN, S. A Goal Programming Procedure for Solving Interval Valued Multiobjective Fractional Programming Problems. 16th International Conference on Advanced Computing and Communications, 2008.

PRASAD, S.; ZAHEERUDDIN; LOBIYAL, D. K. Multiobjective multicast routing in wireless ad hoc networks - An Ant Colony approach, 2013 IEEE 3rd International Advance Computing Conference (IACC), 2013.

RITZEL, B. J.; EHEART, J. W.; RANJITHAN, S. Using genetic algorithms to solve a multiple objective groundwater pollution containment problem. Water Resources Research, v. 30, n. 5, p. 1589–1603, 1994.

RUBIO-LARGO, A.; VEGA-RODRÍGUEZ, M. A.; GONZÁLEZ-ÁLVAREZ, D. L. A Hybrid Multiobjective Memetic Metaheuristic for Multiple Sequence Alignment. IEEE Transactions on Evolutionary Computation, Vol. 20, No. 4, 2016.

SANTANDER-JIMÉNEZ, S. ; VEGA-RODRÍGUEZ, M. A. Applying OpenMP-based parallel implementations of NSGA-II and SPEA2 to study phylogenetic relationships, IEEE International Conference on Cluster Computing (CLUSTER), 2014.

SANTOS, A. C. M. DOS. Aprendizado de máquina aplicado ao diagnóstico de Dengue, XIII Encontro Nacional de Inteligência Artificial e Computacional, Pernambuco, 2016.

SANTOS, A. R. dos. Síntese de Árvores de Padrões Fuzzy através de Programação Genética Cartesiana, Dissertação de mestrado, Universidade do Estado do Rio de Janeiro, 2014.

SANTOS, F. M. DA C.; DA SILVA, I. N.; SUETAKE, M. Sobre a aplicação de sistemas inteligentes para diagnóstico de falhas em máquinas de indução - Uma visão geral, Sba Controle & Automação, v.23, n.5, p.553-569, 2012.

SCHAFFER, J. Multiple objective optimization with vector evaluated genetic algorithms. 1st Int. Conf. on Genetic Algorithms, Pittsburgh, 1985.

SHAHVERDI, M.; MAZZOLA, M. S.; GRICE, Q.; DOUDE, M. Pareto Front of Energy Storage Size and Series HEV Fuel Economy Using Bandwidth-Based Control Strategy. IEEE Transactions on Transportation Electrification, Vol. 2, No. 1, 2016.

SHAMSHIRBAND, S.; SHOJAFAR, M.; ALI, A. R. A Solution for Multi-objective Commodity Vehicle Routing Problem by NSGA-II. 14th International Conference on Hybrid Intelligent Systems (HIS), 2014.

SHI, C. G.; ZHOU, J. J.; WANG, F. Low probability of intercept optimization for radar network based on mutual information, IEEE China Summit & International Conference on Signal and Information Processing, p.683-687, 2014.

SHUKLA, P. K.; TRIPATHI, S. P. A Review on the Interpretability-Accuracy Trade-Off in Evolutionary Multi-Objective Fuzzy Systems (EMOFS), Information, 3, p. 256-277, 2012.

SRINIVAS, N.; DEB, K. Multiobjective optimization using nondominated sorting in genetic algorithms, Evolutionary Computation, vol. 2, pp. 221-248, 1995.

STEUER, R. E.; CHOO, R.-U. An interactive weighted Tchebycheff procedure for multiple objective programming, Mathematical Programming, vol. 26, pp.326- 344, 1983.

STRENGTH PARETO EVOLUTIONARY ALGORITHM 2 IN MATLAB. Disponível em: <http://www.yarpiz.com >. Acesso em: 08 dez. 2016.

TANSCHEIT, R. Sistemas fuzzy, In: Inteligência computacional: aplicada à administração, economia e engenharia em Matlab, pp. 229–264, São Paulo, Thomson Learning, 2007.

VERDEJO, H.; GONZÁLEZ, D.; DELPIANO, J.; BECKER, C. Tuning of Power System Stabilizers using Multiobjective Optimization NSGA II. IEEE LATIN AMERICA TRANSACTIONS, VOL. 13, NO. 8, pp. 2653-2660, 2015.

VERLY, A.; RICCO, R. A.; SANTOS, F. G. DOS; MAZZINI, H. M. Controle aplicado em tempo real a uma planta de temperatura: Resultados experimentais. 9th IEEE/las International Conference on Industry Applications – INDUSCON, 2010.

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

ZADEH, L. A. Fuzzy sets. Information and Control, vol. 8, pp. 338-353, 1965.

ZEBULUM, R. S. Síntese de Circuitos Eletrônicos por Computação Evolutiva. Tese de Doutorado. Departamento de Engenharia Elétrica da Pontifícia Universidade Católica do Rio de Janeiro, 1999.

ZHANG, Q.; LI, H., MOEA/D: A Multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation, vol. 11, no. 6, pp. 712–731, 2007.

ZHU, W.; YASEEN, A.; LI, Y. DEMCMC-GPU: An Efficient Multi- Objective Optimization Method with GPU Acceleration on the Fermi Architecture. New Generation Computing, 29(2):163–184, 2011.

ZIEGLER, J. G.; NICHOLS, N. B. Optimum settings for automatic controllers. Transactions of the A. S. M. E., pp. 759-768, 1942.

ZITZLER, E.; DEB, K.; THIELE, L. Comparison of multiobjective evolutionary algorithms: Empirical results, Evol. Comput., Vol. 8, No. 2, pp. 173–195, 2000.

ZITZLER, E.; LAUMANNS, M.; THIELE, L. SPEA2: Improving the strength pareto evolutionary algorithm. Technical report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), 2001.

ZITZLER, E.; THIELE, L., Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach, IEEE Transaction on evolutionary computation, vol. 3, p. 257-271, 1999.