UNIVERSIDADE FEDERAL DE MINAS GERAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ALGORITMOS EVOLUCIONÁRIOS INTERVALARES PARA OTIMIZAÇÃO ROBUSTA MULTIOBJETIVO

Marcus Henrique Soares Mendes

Tese submetida à banca examinadora designada pelo colegiado do programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, como requisito parcial à obtenção de título de Doutor em Engenharia Elétrica.

Orientador: Prof. Dr. João Antônio de Vasconcelos (DEE/UFMG)
Coorientador: Prof. Dr. Gustavo Luís Soares (PPGEE/PUC-MG)

Linha de Pesquisa: Otimização

Belo Horizonte
Minas Gerais - Brasil
Fevereiro - 2013

Ela se estende vigorosamente de um extremo ao outro, e governa retamente o universo.

A companheira ideal

Amei a sabedoria e a busquei desde minha juventude, e procurei tomá-la como esposa, pois fiquei enamorado de sua formosura. A união com Deus manifesta a nobre origem dela, porque o Senhor do universo a amou. De fato, ela é iniciada na ciência de Deus e seleciona as obras dele. Se na vida a riqueza é um bem desejável, que riqueza é maior do que a sabedoria, que tudo produz? E se é a inteligência que opera, quem mais que ela é artífice do que existe? Se alguém ama a justiça, as virtudes são seus frutos, pois é ela quem ensina a temperança e a prudência, a justiça e a fortaleza, que são na vida os bens mais úteis aos homens. Se alguém deseja também uma rica experiência, ela conhece as coisas passadas e entrevê as futuras, conhece as sutilezas das oratórias e a solução dos enigmas. Prevê sinais e prodígios, e o desenrolar das épocas e tempos. Decidi, portanto, tomá-la por companheira de minha vida, sabendo que ela será para mim uma boa conselheira e me trará conforto nas preocupações e no sofrimento. Por causa dela, serei elogiado pelas assembleias e, ainda jovem, os anciãos me honrarão. No julgamento, terei agudeza e serei admirado pelos poderosos. Se eu me calar, ficarão na expectativa; se eu falar, me prestarão atenção; se eu prolongar o discurso, levarão a mão à boca. Por meio dela alcançarei a imortalidade, e deixarei à posteridade uma lembrança eterna. Governarei os povos e dominarei as nações. Tiranos terríveis ficarão assustados quando me ouvirem; serei bom para com o povo e corajoso no combate. Ao voltar para casa, repousarei junto dela, porque a sua companhia não provoca amargura, e a convivência com ela não traz nenhuma dor, pelo contrário, traz contentamento e alegria!

(Livro da Sabedoria 8, 1-16)

Agradecimentos

A Deus por ter me dado saúde e discernimento para realizar com sucesso as atividades desta tese.

Aos meus pais por terem se sacrificado para permitir que eu pudesse ir cada vez mais longe e por terem me ensinado que independente de conquistas, fracassos, distâncias, alegrias e tristezas, o que realmente importa é o amor e carinho que há em nossos corações.

À Taís por ser uma companheira dedicada, amorosa e por não ter perdido a paciência comigo pelas inúmeras vezes que a pedi para ler e reler meus textos e artigos. Apesar de não entender nada de algoritmos, ela ouvia atentamente minhas ideias e me ajudava a encontrar a solução dos problemas.

Ao professor João Vasconcelos pelo tempo investido na orientação desta tese, pelos ensinamentos recebidos, pela atenção, exigência, paciência, prontidão, pelas críticas construtivas, por proporcionar-me parcerias no exterior e ensinar-me como ser um pesquisador.

Ao professor Gustavo Soares pela disponibilidade, pelo zelo, conselhos e palavras de motivação.

Aos professores Pedro Peres, Petr Ekel e Carlos Maia pela participação e pelas contribuições realizadas no exame de qualificação.

Ao professor Jean-Louis Coulomb pela contribuição em cinco artigos provenientes desta tese.

Ao professor Carlos Fonseca pela receptividade, atenção, disponibilidade, sugestões e supervisão do meu intercâmbio acadêmico na Universidade de Coimbra em Portugal.

À pesquisadora Sara Silva por ter me recebido no INESC-ID em Lisboa e ter compartilhado sua experiência sobre programação genética.

A minha irmã Glauce, pela ajuda no inglês e pelo apoio recebido.

Aos meus familiares pelo apoio e votos de sucesso recebidos.

Aos colegas do Laboratório de Computação Evolucionária, em especial, Carlos, João Batista, Marconi, Rafael, Claret e Moisés pela amizade, troca de ideias e pelas angústias compartilhadas.

A todos os meus amigos da UFV, da UFMG, de JF, de Florestal, de BH, da Pedro Nunes pela amizade, companheirismo e convivência.

Às secretárias do PPGEE da UFMG, Anete Vieira e Arlete de Freitas, pelo auxílio nas questões formais do programa.

A UFV por permitir meu afastamento total durante boa parte do doutoramento.

Às servidoras da PPG da UFV, Suely e Margarida, pela prontidão no atendimento de todas as minhas solicitações durante o período de afastamento para treinamento.

Aos funcionários da UFV – Campus Florestal pelo suporte durante o período de afastamento, em especial, aos professores Calil e Marco Antônio e aos servidores José Aparecido e Maria Lúcia.

Aos revisores anônimos pelas sugestões recebidas.

A CAPES por financiar parte desta pesquisa.

Resumo

Os problemas reais de otimização multiobjetivo podem estar sujeitos a incertezas, as quais muitas vezes são impossíveis de serem evitadas. Com isso, há possibilidade de que uma pequena incerteza faça com que uma solução numérica ótima obtida para um problema real torne-se completamente sem sentido na prática. Assim, o escopo do processo de otimização multiobjetivo amplia-se e requer metodologias capazes de obter soluções robustas, ou seja, que funcionem perfeitamente em ambientes incertos. Nesta tese, são propostos algoritmos evolucionários intervalares que visam buscar soluções robustas para problemas de otimização multiobjetivo. Como noção de robustez, considera-se o cenário de pior caso das incertezas relacionadas às variáveis de decisão, aos parâmetros do ambiente e aos ruídos nas funções objetivo. O tratamento das incertezas é feito pela análise intervalar. Duas formulações matemáticas do problema de otimização robusta multiobjetivo relativas à noção robusta de cenário de pior caso são consideradas: *minimax* e *minimax regret*. Para lidar com tais formulações são propostos os métodos IRMOEA-M e IRMOEA-MR, respectivamente. Ambos os métodos são descritos detalhadamente e têm seu desempenho verificado perante um conjunto de problemas testes e reais. Além dessas contribuições originais, busca-se lidar com os principais fatores complicadores que surgem mediante o emprego da análise intervalar para lidar com as incertezas: (I) Dificuldades em obter funções de inclusão e (II) Possibilidade da imagem das soluções robustas não pertencer à imagem viável. Para lidar com o fator complicador I, os métodos SNIF-GPA e SNIF-MOGPA, baseados em programação genética são propostos. Em relação ao fator complicador II, define-se a fronteira ideal de maximização. A fim de avaliar o desempenho dos métodos propostos realizaram-se experimentos computacionais envolvendo funções teste e problemas reais nas áreas de eletromagnetismo e engenharia de controle. Os resultados obtidos indicaram que os métodos baseados em programação genética foram capazes de obter boas funções de inclusão, inclusive, quando comparados a outras metodologias. Adicionalmente, a fronteira ideal de maximização mostrou-se promissora e competitiva quando usada em um método robusto estritamente intervalar e em um evolucionário intervalar.

Abstract

The real-world multi-objective optimization problems may be subjected to uncertainties which are often impossible to be avoided in practice. Hence, there is the possibility that a small uncertainty becomes a numerical optimal solution obtained for a real problem completely meaningless in practice. Thereby, the scope of the multi-objective optimization process expands and requires methodologies capable of obtaining robust solutions, namely that operate perfectly in uncertain environments. In this thesis, interval evolutionary algorithms are proposed to find robust solutions to multi-objective optimization problems. The considered notion of robustness is the worst-case scenario of uncertainties related to the decision variables, the environmental parameters, and the noise in the objective functions. The treatment of uncertainties is performed by interval analysis. Two mathematical formulations of robust multi-objective optimization problem related to the robust notion of worst-case scenario are considered: *minimax* and *minimax regret*. To deal with such formulations, the methods IRMOEA-M and IRMOEA-MR are proposed, respectively. Both methods are described in detail and have their performance evaluated against a set of test and real problems. Besides these original contributions, we cope with the major complicating factors that arise through the use of interval analysis to deal with uncertainties: (I) Difficulties in obtaining inclusion functions, and (II) Possibility of image of robust solutions does not belong to feasible image. To deal with the complicating factor I, the methods SNIF-GPA and SNIF-MOGPA, based on genetic programming are proposed. Regarding the complicating factor II, the ideal frontier of maximization is defined. Computational experiments involving test functions and real problems in the fields of electromagnetic and control engineering were performed to evaluate the performance of the proposed methods. The results indicated that methods based on genetic programming were able to obtain good inclusion functions even when compared to other methodologies. Additionally, the ideal frontier of maximizing was promising and competitive when utilized in a strictly interval robust method and in an interval evolutionary robust method.

CONTRIBUIÇÕES

As principais contribuições geradas ao longo do desenvolvimento desta tese são listadas nos tópicos a seguir.

Publicações

[2013] **Mendes, M. H. S**., Soares, G. L., Coulomb, J. L., and Vasconcelos, J. A., "A Surrogate Genetic Programming Based Model to Facilitate Robust Multi-Objective Optimization: A Case Study in Magnetostatics," **IEEE Transactions on Magnetics** (aceito em 31/12/2012, DOI: 10.1109/TMAG.2013.2238615).

[2013] **Mendes, M. H. S**., Soares, G. L., Coulomb, J. L., and Vasconcelos, J. A., "Appraisal of Surrogate Modeling Techniques: A Case Study of Electromagnetic Device," **IEEE Transactions on Magnetics** (aceito em 15/01/2013, DOI: 10.1109/TMAG.2013.2241401).

[2012] **Mendes, M. H. S**., Soares, G. L., Coulomb, J. L., and Vasconcelos, J. A., "**Surrogate Model Determination by Using Genetic Programming**," in 15th Biennial IEEE Conference on Electromagnetic Field Computation - CEFC 2012, Oita, Japan, November, 2012.

[2012] **Mendes, M. H. S**., Soares, G. L., Coulomb, J. L., and Vasconcelos, J. A., "**Appraisal of Surrogate Modeling Techniques for Electromagnetic Device**," in 15th Biennial IEEE Conference on Electromagnetic Field Computation - CEFC 2012, Oita, Japan, November, 2012.

[2012] **Mendes, M. H. S**., Soares, G. L., Coulomb, J. L., and Vasconcelos, J. A., "**Comparison of Surrogate Modeling Approaches on TEAM Workshop Problem 22,**" in 15º SBMO – Simpósio Brasileiro de Micro-ondas e Optoeletrônica e o 10º CBMag – Congresso Brasileiro de Eletromagnetismo - MOMAG 2012, João Pessoa, Brasil, Agosto, 2012.

[2010] **Mendes, M. H. S**., Soares, G. L., and Vasconcelos, J. A., "PID Step Response Using Genetic Programming," **Lecture Notes in Computer Science**, Vol. 6457, pp. 359-368, 2010.

[2010] **Mendes, M. H. S**., Cosme, L. B., Caminhas, W. M., and Vasconcelos, J. A., "Aplicação de Modelagem Nebulosa em Problemas de Otimização sob Condições de Incertezas," in 1º Congresso Brasileiro de Sistemas Fuzzy – CBSF-2010, pp. 463-470, Sorocaba, Brasil, Novembro, 2010.

Conceitos e Definições

Os principais conceitos e definições propostos são: *surrogate natural inclusion function* (Seção 7.1), fronteira ideal de maximização (Definição 7-3 da Seção 7.2) e operador de *regret* no caso multiobjetivo (Equações (2-13) e (7-9)).

Algoritmos

- SNIF-GPA (*Surrogate Natural Inclusion Function by Genetic Programming Algorithm*), detalhado na Seção 7.1.
- SNIF-MOGPA (*Surrogate Natural Inclusion Function by Multi-Objective Genetic Programming Algorithm*), detalhado na Seção 7.1.
- IRMOEA-M (*Interval Robust Multi-Objective Evolutionary Algorithm – Minimax*), detalhado na Seção 7.3.
- IRMOEA-MR (*Interval Robust Multi-Objective Evolutionary Algorithm – Minimax Regret*), detalhado na Seção 7.4.
- IREA-M (*Interval Robust Evolutionary Algorithm – Minimax*), detalhado na Seção 7.4.

Outras Contribuições

Esta tese contribui para o Programa de Pós-Graduação em Engenharia Elétrica (PPGEE) da Universidade Federal de Minas Gerais por meio do avanço em duas linhas de pesquisa recém-investigadas no âmbito do PPGEE. São elas: Otimização Robusta Multiobjetivo e Análise Intervalar.

Lista de Abreviaturas

ASU	*Attainment Surface*
EAFs	*Empirical Attainment Functions*
EAs	*Evolutionary Algorithms* – Algoritmos Evolucionários
EC	*Evolutionary Computation* – Computação Evolucionária
FPO	Fronteira Pareto Ótima
GAs	*Genetic Algorithms* – Algoritmos Genéticos
GP	*Genetic Programming* – Programação Genética
IA	*Interval Analysis* – Análise Intervalar
IREA-M	*Interval Robust Evolutionary Algorithm – Minimax*
[I]RMOA I	*Interval Robust Multi-Objective Algorithm I*
[I]RMOA II	*Interval Robust Multi-Objective Algorithm II*
[I]RMOEA	*Interval Robust Multi-Objective Evolutionary Algorithm*
IRMOEA-M	*Interval Robust Multi-Objective Evolutionary Algorithm – Minimax*
IRMOEA-MR	*Interval Robust Multi-Objective Evolutionary Algorithm – Minimax Regret*
MFO	Melhor Fronteira Obtida
µGA	*Micro Genetic Algorithm*
MOEAs	*Multi-Objective Evolutionary Algorithms* – Algoritmos Evolucionários Multiobjetivo
MOGA	*Multi-Objective Genetic Algorithm* – Algoritmo Genético Multiobjetivo
MOPs	*Multi-Objective Optimization Problems* – Problemas de Otimização Multiobjetivo
MSE	*Mean Squared Error*
NSGA	*Nondominated Sorting Genetic Algorithm*
NPGA	*Niched-Pareto Genetic Algorithm*
PAES	*Pareto Archived Evolution Strategy*
PID	Proporcional-Integral-Derivativo
RMOP	*Robust Multi-Objective Optimization Problem* – Problema de Otimização Robusta Multiobjetivo
SPEA	*Strength Pareto Evolutionary Algorithm*
SR	*Symbolic Regression* – Regressão Simbólica
VEGA	*Vector Evaluated Genetic Algorithm*

Caso a abreviatura tenha versão comumente utilizada na língua inglesa, por questões de padronização, optou-se por utilizá-la.

Lista de Símbolos

$\mathbb{R}^n$	Espaço Euclidiano de n dimensões
$\mathbb{R}$	Conjunto de todos os números reais
$\mathbb{IR}$	Conjunto de todos os números reais intervalares
$\emptyset$	Conjunto vazio
$\mathbb{B}$	Conjunto dos números booleanos
$\mathbb{IB}$	Conjunto dos números intervalares booleanos
$\mathbf{x}$	Vetor das variáveis de decisão
$\boldsymbol{f}(\mathbf{x})$	Vetor de funções objetivo
$\mathbf{X}$	Espaço de busca - conjunto de vetores das variáveis de decisão
$\mathbf{S}$	Espaço de busca viável
$\mathbf{Z}$	Imagem do espaço de busca viável
$\mathbf{P}$	Espaço das incertezas
n_f	Dimensão do espaço dos objetivos
n_v	Dimensão do espaço de busca
n_p	Dimensão do espaço de incerteza
n_g	Quantidade de restrições de desigualdde
n_h	Quantidade de restrições de igualdade
n_s	Quantidade de caixas no subpavimento do espaco das incertezas
$f_i(\mathbf{x})$	Função objetivo índice i
$g_i(\mathbf{x})$	Restrição de desigualdade índice i
$h_j(\mathbf{x})$	Restrição de igualdade índice j
$[\mathrm{x}] = [\mathrm{x}^-, \mathrm{x}^+]$	Escalar intervalar
$[\mathbf{x}] = [\mathbf{x}^-, \mathbf{x}^+]$	Vetor intervalar (caixa)
f	Função real
f([x])	Função intervalar
[f]([x])	Função de inclusão
$\mathrm{x}^- = lb([\mathrm{x}])$	Limite inferior de [x]
$\mathrm{x}^+ = ub([\mathrm{x}])$	Limite superior de [x]
$w([\mathrm{x}])$	Largura de [x]
$mid([\mathrm{x}])$	Centro de [x]
$\boldsymbol{RSP}[\mathbf{x}]$	Subpavimento regular de $[\mathbf{x}]$
$\mathbf{F}^{\max}$	Fronteira ideal de maximização
$\mathbf{u}^{\max}$	Ponto ideal de maximização

Caractere minúsculo e em negrito indica que é um vetor. Caractere maiúsculo e em negrito é usado para conjunto, com exceção dos primeiros seis conjuntos listados acima.

Lista de Figuras

Figura 2-1 – Exemplo do espaço das variáveis de decisão (n=2) e do espaço dos objetivos (k=2). 9
Figura 2-2 – Relações de dominância em relação ao ponto cinza. 10
Figura 2-3 – Os conjuntos **S**, **Z** e a fronteira Pareto ótima. 10
Figura 4-1 – Exemplo das Relações de dominância entre os vetores no espaço dos objetivos..... 28
Figura 4-2 - Fronteira Pareto ótima **P** e os conjuntos de soluções não-dominadas $\mathbf{A}_1$, $\mathbf{A}_2$ e $\mathbf{A}_3$ de um MOP. 29
Figura 4-3 – Representação de um indivíduo utilizando a estrutura de árvore. 31
Figura 4-4 – Exemplo de cruzamento. 34
Figura 4-5 – Exemplo de mutação. 34
Figura 5-1 - Operações sobre conjuntos. 37
Figura 5-2- caixa [**x**] em $\mathbb{IR}2$. 39
Figura 5-3- Imagem de uma caixa pela função **f** e duas de suas funções de inclusão [**f**] e [**f**]*. Sendo [**f**]* mínima. Adaptada de (SOARES, 2008). 43
Figura 6-1-Pontos de referência e pontos encontrados pela expressão de ajuste no: A) cenário 1 e B) cenário 2. 48
Figura 7-1 – Estrutura geral do nó da árvore. 51
Figura 7-2 – Estrutura de nós para a expressão $\sin x + x + x$. 52
Figura 7-3 – Ambas as árvores têm três nós. Entretanto, o somatório do número de nós de todas as subárvores é diferente: para a árvore A é 5 = 3 + 1 + 1 e para a árvore B é 6 = 3 + 2 + 1. 56
Figura 7-4 – Apesar das duas árvores serem equivalentes a $x2 + 4x + 4$, a da esquerda tem complexidade de 11 = 5+3+1+1+1, a da direita tem complexidade de 37 = 11+9+3+1+1+5+3+1+1+1+1. Neste caso, o vetor objetivo da primeira expressão domina o da segunda..... 56
Figura 7-5-Cálculo de **M**, **u**max e **F**max de um ponto viável **x** sujeito às incertezas em **P**. 59
Figura 7-6-Obtenção da fronteira não-dominada para os pontos **a, b** e **c** considerando-se **F**max e **u**max na computação das incertezas em **P**. 61
Figura 7-7-Obtenção dos pontos de referência para o subpavimento de **P** no IRMOEA-MR. 67
Figura 8-1 - Características da Resposta ao Degrau Unitário. Adaptada de (OGATA, 1997)..... 68
Figura 8-2 – Soluções dominadas e não-dominadas para dez execuções do algoritmo em ambos os esquemas, considerando cada uma das características de resposta ao degrau como segue: a) ITAE, b) Sobre-sinal, c) Tempo de pico, d) Tempo de subida e e) Tempo de acomodação. 74
Figura 8-3 – Fronteira robusta do problema (8-13) obtida pelo [I]RMOA-II utilizando-se **u**max com $\varepsilon x = 0{,}6$ e $\varepsilon p = 0{,}05$..... 79
Figura 8-4 – Soluções robustas no espaço das variáveis de decisão. 79
Figura 8-5 – Resposta ao degrau unitário das 617 soluções robustas. 80
Figura 8-6 – Espaço dos objetivos do problema (8-24). 83
Figura 8-7 – Espaço das variáveis de decisão do problema (8-24). 83
Figura 8-8 – Resposta do sistema ao degrau unitário de acordo com as soluções da Figura 8-6 (B). 84
Figura 8-9 – Problema TEAM 22 com incertezas (SOARES *et al.*, 2009a). 85

Figura 8-10 – Metodologia proposta para aplicação do SNIF-GPA ao problema TEAM 22. 86
Figura 8-11 – Comparação entre as fronteiras robustas. 92
Figura 8-12 – Importância das soluções robustas no problema TEAM 22, extraído de (SOARES *et al.*, 2009a). 94
Figura 8-13 – Imagem viável e fronteira robusta do problema teste I utilizando **u**max. 97
Figura 8-14 – Fronteira robusta do problema teste I utilizando **F**max 98
Figura 8-15 – Fronteira robusta do problema teste I utilizando **u**max (círculos brancos) e **F**max (asteriscos) com $\varepsilon x = 0{,}1$. 98
Figura 8-16 – Fronteira robusta do problema SCH2 utilizando **u**max (asteriscos em cinza escuro) e **F**max (asteriscos pretos) com $\varepsilon x = 0{,}1$. 99
Figura 8-17 – Fronteira robusta do problema FON utilizando **u**max (círculos) e **F**max (pontos em azul) com $\varepsilon x = 0{,}1$. 100
Figura 8-18 – Fronteira robusta do problema ZDT2 utilizando **u**max (círculos) e **F**max (pontos em cinza escuro) com $\varepsilon x = 0{,}1$. 101
Figura 8-19 – ASUs para o problema SCH2 na formulação robusta *minimax*. 102
Figura 8-20 – ASUs para o problema FON na formulação robusta *minimax*. 102
Figura 8-21 – Mediana ASUs para o problema FON na formulação robusta *minimax*. 103
Figura 8-22 – ASUs para o problema ZDT2 na formulação robusta *minimax*. 103
Figura 8-23 – Mediana ASUs para o problema ZDT2 na formulação robusta *minimax*. 104
Figura 8-24 – Viga em seção I (COELLO e CHRISTIANSEN, 1998). 104
Figura 8-25 – ASUs para o problema da viga na formulação robusta *minimax*. 105
Figura 8-26 – ASUs para o problema SCH2 na formulação robusta *minimax regret*. 106
Figura 8-27 – ASUs para o problema FON na formulação robusta *minimax regret*. 107
Figura 8-28 – ASUs para o problema ZDT2 na formulação robusta *minimax regret*. 107
Figura 8-29 – ASUs para o problema da viga na formulação robusta *minimax regret*. 108

Lista de Tabelas

Tabela 1-1- Principais contribuições desta tese. 6
Tabela 3-1- Perfil dos principais trabalhos consultados na revisão sobre otimizacão robusta. 23
Tabela 4-1– Principais técnicas utilizadas pelos MOEAs para atender a cada meta. 26
Tabela 4-2- Relações de dominância entre os vetores no espaço dos objetivos. 27
Tabela 4-3 - Relações de dominância entre conjuntos de soluções não-dominadas no espaço dos objetivos. 28
Tabela 6-1 - Amostra de pontos. 46
Tabela 8-1 - Valores dos parâmetros físicos para o problema de posição do motor de corrente contínua. 70
Tabela 8-2 – Parâmetros do SNIF-GPA. 70
Tabela 8-3 – MSE da melhor expressão obtida para cada característica na fase de treinamento. 71
Tabela 8-4 – Tempo de execução necessário pelo SNIF-GPA para a obtenção da melhor expressão em cada característica. 71
Tabela 8-5 – MSE da melhor expressão obtida para cada característica na fase de validação. 72
Tabela 8-6 – Tolerância da melhor expressão obtida para cada característica na fase de validação. 72
Tabela 8-7 – Inclinação e intercepto para as melhores expressões obtidas. 73
Tabela 8-8 – Parâmetros do SNIF-MOGPA. 74
Tabela 8-9 – Inclinação e intercepto para as expressões do ITAE e tempo de subida exibidas. 77
Tabela 8-10 – Parâmetros do algoritmo SNIF-GPA. 81
Tabela 8-11 – Média do erro quadrático médio (MSE) para cada característica em dez execuções do SNIF-GPA. 82
Tabela 8-12 – Configuração do SNIF-GPA. 86
Tabela 8-13 – Resultados do SNIF-GPA para F_1. 87
Tabela 8-14 – Resultados do SNIF-GPA para F_2. 87
Tabela 8-15 – Resultados do SNIF-GPA para C_1. 87
Tabela 8-16 – Resultado da RBF-NN para o problema TEAM 22. 89
Tabela 8-17 – Resultado do *Universal* Kriging para o problema TEAM 22. 90
Tabela 8-18 – Resultado para o critério de adequação à otimização robusta intervalar. 90
Tabela 8-19 – Comparação do custo computacional. 93
Tabela 8-20 – Parâmetros do SNIF-MOGPA aplicado ao problema desafio. 96
Tabela 8-21 – Resultados de dez execuções do SNIF-MOGPA para o problema desafio. 96
Tabela 8-22 – Resultados ANN-RBF para o problema desafio. 97
Tabela 8-23 – Valor de εp para cada problema. 101

Lista de Algoritmos

Algoritmo 4-1– Estrutura geral dos algoritmos evolucionários. 25
Algoritmo 4-2– Estrutura geral de um MOEA. 26
Algoritmo 4-3– Algoritmo de programação genética. 30
Algoritmo 7-1– Fase de treinamento do SNIF-GPA. 53
Algoritmo 7-2– Conversão da árvore em uma expressão. 53
Algoritmo 7-3– Fase de validação do SNIF-GPA. 55
Algoritmo 7-4– Fase de treinamento do SNIF-MOGPA. 57
Algoritmo 7-5– Fase de validação do SNIF-MOGPA. 57
Algoritmo 7-6– Geração do subpavimento para o espaço de incertezas. 58
Algoritmo 7-7– Computação do conjunto $\mathbf{M} = \mathbf{fx}, \mathbf{P} +$. 61
Algoritmo 7-8– Computação de $\mathbf{u}\max\mathbf{M}$. 61
Algoritmo 7-9– Computação de $\mathbf{F}\max(\mathbf{M})$. 61
Algoritmo 7-10– Tratamento de restrições para o indivíduo $\mathbf{x}$. 63
Algoritmo 7-11– IRMOEA-M. 64
Algoritmo 7-12– IRMOEA-MR. 66
Algoritmo 7-13– IREA-M. 66

Sumário

1. INTRODUÇÃO 1
1.1 Justificativa 2
1.2 Objetivos 5
1.2.1 Objetivo Geral 5
1.2.2 Objetivos Específicos 5
1.3 Delimitações 5
1.4 Contribuições 6
1.5 Organização do Trabalho 6
1.6 Conclusão 7
2. OTIMIZAÇÃO MULTIOBJETIVO 8
2.1 O Problema de Otimização Robusta Multiobjetivo 10
2.1.1 Formulação Matemática do RMOP pela Noção Robusta do *minimax* 11
2.1.2 Formulação Matemática do RMOP pela Noção Robusta do *minimax regret* 11
2.2 Conclusão 12
3. INCERTEZA E ROBUSTEZ 13
3.1 Otimização Robusta 17
3.1.1 Otimização Robusta Baseada na Função *fitness* Esperada 18
3.1.2 Otimização Robusta Baseada no Cenário de Pior Caso 19
3.2 Conclusão 24
4. COMPUTAÇÃO EVOLUCIONÁRIA 25
4.1 Fundamentos 25
4.2 Algoritmos Evolucionários Multiobjetivo 26
4.3 Comparação de Desempenho dos MOEAs 27
4.4 Programação Genética 29
4.4.1 Fundamentos 29
4.4.2 Função de *Fitness* 32
4.4.3 Operações Genéticas 33
4.5 Conclusão 34
5. ANÁLISE INTERVALAR 36
5.1 Noções da Teoria de Conjuntos 37
5.2 Intervalos – Conceitos Básicos 38
5.2.1 Vetor Intervalar 39
5.3 Computação Intervalar 40
5.3.1 Operações Aritméticas 40
5.3.2 Funções Intervalares 41
5.3.3 Funções de Inclusão 42
5.3.4 Problema da Dependência 44

5.3.5 Testes de Inclusão 44
5.4 Subpavimentos 45
5.5 Conclusão 45
6. REGRESSÃO SIMBÓLICA 46
6.1 Exemplo de Regressão Simbólica 46
6.2 Análise Intervalar na Regressão Simbólica 49
6.3 Regressão Simbólica com *Linear Scaling* 49
6.4 Conclusão 50
7. ALGORITMOS E CONCEITOS PROPOSTOS 51
7.1 Aproximação de Funções de Inclusão Via Programação Genética 51
7.1.1 Estrutura dos Indivíduos no SNIF-GPA e SNIF-MOGPA 51
7.1.2 Estrutura Geral do SNIF-GPA 52
7.1.3 Estrutura Geral do SNIF-MOGPA 55
7.2 Maximização da Influência das Incertezas no Contexto Intervalar 57
7.3 *Interval Robust Multi-Objective Evolutionary Algorithm – Minimax* 62
7.3.1 Estrutura Geral do IRMOEA-M 62
7.4 *Interval Robust Multi-Objective Evolutionary Algorithm – Minimax Regret* 64
7.5 Conclusão 67
8. EXPERIMENTOS E RESULTADOS 68
8.1 Aproximação de Funções de Inclusão em Problemas de Engenharia de Controle 68
8.1.1 Aplicação do SNIF-GPA ao Problema do Motor de Corrente Contínua 70
8.1.2 Aplicação do SNIF-MOGPA ao Problema do Motor de Corrente Contínua 73
8.1.3 Aplicação do [I]RMOA-II para Sintonia Robusta do Controlador PID no Problema do Motor de Corrente Contínua 77
8.1.4 Aplicação do SNIF-GPA ao Problema do Controlador PD 80
8.1.5 Aplicação do [I]RMOA-II para Sintonia Robusta do Controlador PD 82
8.2 Aproximação de Funções de Inclusão em Eletromagnetismo 84
8.2.1 Aplicação do SNIF-GPA ao Problema TEAM 22 85
8.2.2 Análise Comparativa do SNIF-GPA no Problema TEAM 22 com Incertezas 88
8.2.3 Aplicação do IRMOEA-M ao Problema TEAM 22 com Incertezas 90
8.3 Aplicação do SNIF-MOGPA ao Problema Desafio 94
8.4 Fronteira Ideal de Maximização 97
8.5 Aplicação do IRMOEA-M 101
8.6 Aplicação do IRMOEA-MR 105
8.7 Conclusão 108
9. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS 110
9.1 Conclusões 110
9.2 Trabalhos Futuros 111
REFERÊNCIAS BIBLIOGRÁFICAS 112

1. INTRODUÇÃO

> To be uncertain is to be uncomfortable, but to be certain is to be ridiculous.
>
> Provérbio Chinês.

Várias questões relacionadas à sociedade moderna envolvem a solução de problemas de otimização com múltiplos e conflitantes objetivos, utilizados na construção do modelo matemático de problemas reais, cujas soluções permitem ao decisor escolher aquela de melhor qualidade segundo suas preferências. Dessa forma, no mundo real é comum a existência de problemas de otimização multiobjetivo (*Multi-Objective Optimization Problems* – MOPs) em várias áreas do conhecimento, sobretudo na engenharia. Por se tratarem de problemas complexos, é necessário o uso de ferramentas poderosas de otimização. Uma das ferramentas que vem sendo bastante utilizada é a computação evolucionária (*Evolutionary Computation* - EC), a qual é baseada na teoria da evolução natural. A EC reúne métodos estocásticos com boa facilidade de uso, alta aplicabilidade às mais variadas classes de problemas e capazes de lidar com problemas descontínuos, multimodais, com ótimos isolados e/ou variáveis de decisão mistas. Além disso, enquanto os algoritmos de otimização determinísticos trabalham com uma única solução no processo de busca, os algoritmos evolucionários (*Evolutionary Algorithms* - EAs) trabalham com uma população de soluções candidatas. Isso permite em uma única execução encontrar um conjunto de soluções, o que é extremamente útil em MOPs (DIAS e VASCONCELOS, 2002; COELLO *et al.*, 2007).

Os problemas reais de otimização, além do caráter multiobjetivo, também estão sujeitos a incertezas, as quais muitas vezes são difíceis ou impossíveis de serem evitadas. As incertezas são comumente associadas à imprecisão nas medidas das variáveis de decisão, mudanças nas condições do ambiente (e. g. alterações na temperatura, pressão, velocidade, entre outros) e ruído na função *fitness*, em geral, proveniente de erros de simulação. Com isso, há possibilidade de que uma pequena incerteza faça com que a solução nominal[1] ótima obtida para um problema real se torne completamente sem sentido na prática (BEN-TAL *et al.*, 2009). Seja pelo fato da ocorrência da incerteza gerar uma variação considerável de desempenho no sistema otimizado e/ou por

[1] Solução que não considera as incertezas.

comprometer a factibilidade da solução (LEE e PARK, 2001). Sendo assim, o escopo do processo de otimização multiobjetivo torna-se mais amplo e requer metodologias capazes de obter soluções que funcionem adequadamente em ambientes incertos. Devido aos desafios e à importância prática, esse tópico de pesquisa tem recebido muita atenção nos últimos anos. Prova disso, são as sessões em congressos e edições especiais em periódicos científicos a ele dedicadas (YANG *et al.*, 2007).

Neste trabalho, enfocam-se as perturbações relacionadas às *variáveis de decisão*, aos *parâmetros do ambiente* e aos *ruídos* nas funções objetivo. As duas primeiras perturbações são chamadas de incertezas *paramétricas*. Nessa conjuntura o MOP é renomeado para problema de otimização robusta multiobjetivo (*Robust Multi-Objective Optimization Problem*- RMOP). A resolução do RMOP visa encontrar soluções que, apesar de sujeitas à atuação de pequenas perturbações, mantenham-se boas, em termos de desempenho, e factíveis para todo o nível de incerteza especificado. Tais soluções são denominadas *soluções robustas*. É importante frisar que na literatura existem distintos conceitos de robustez, o que será discutido mais adiante em seção específica.

Diante desse contexto é propício estender a aplicabilidade dos EAs ao RMOP, o que pode ser feito por meio da incorporação de mecanismos capazes de lidar com as incertezas.

1.1 Justificativa

É importante salientar que, atualmente, os profissionais envolvidos com problemas reais se preocupam em obter soluções robustas. Suponha a tarefa de construir uma ponte ferroviária. Em problemas dessa natureza, de acordo com BEN-TAL *et al.* (2009), os engenheiros costumam aumentar a margem de segurança relacionada aos parâmetros de projeto, como a espessura das barras, objetivando tratar as incertezas.

Segundo PAENKE *et al.* (2006), há outras situações reais em que obter soluções nominais de alta qualidade não é suficiente, pois devido a possibilidade de atuação de incertezas as soluções obtidas também devem ser robustas. A seguir, são apresentados alguns casos: I) Em projetos de manufatura, geralmente é impossível produzir um item exatamente com as especificações de projeto. Para contornar isso, utilizam-se tolerâncias. II) Em problemas de *scheduling* é importante que a solução seja capaz de tolerar pequenos desvios nos tempos de processamento estimados, além disso, deve ser ajustável em caso de pane nas máquinas. III) Em projeto de circuitos, estes devem ser capazes de funcionar em diferentes condições ambientais, por exemplo, em um dado intervalo de temperatura. IV) No projeto de pás de turbinas, as turbinas devem funcionar em uma série de condições, por exemplo, em diferentes velocidades.

Na área de saúde, PFLUGFELDER *et al.* (2008) propõem um método para contabilizar as incertezas de modo a obter planos de tratamento robustos para a terapia de prótons de intensidade modulada, que é geralmente utilizada no tratamento de câncer.

Nota-se que em várias áreas do conhecimento têm-se necessidade de utilizar métodos capazes de lidar com as incertezas e que gerem boas soluções robustas. Sendo assim, é relevante pesquisar esses tipos de métodos no contexto da otimização.

Dentre os mecanismos utilizados para modelar as incertezas, do ponto de vista computacional, têm-se intervalos, funções de pertinência e funções de densidade de probabilidade (ZANG *et al.*, 2002). A ordem em que foram listados representa o grau de detalhamento (ascendentemente) das incertezas no mecanismo. Portanto, os intervalos simplesmente expressam a faixa de abrangência das variáveis incertas. As funções de pertinência, usadas nas abordagens nebulosas, apresentam um nível intermediário de detalhe. Por fim, as funções de densidade de probabilidade fornecem uma descrição mais refinada dos parâmetros incertos. No âmbito desta pesquisa, optou-se por utilizar a abordagem de intervalos, ou seja, somente são conhecidos os limites dos parâmetros incertos, sem nenhum tipo de detalhamento.

Utilizando-se análise intervalar (*Interval Analysis* - IA) pode-se desenvolver algoritmos para encontrar intervalos que contenham com absoluta certeza a resposta exata para vários problemas matemáticos (MOORE *et al.*, 2009). Nessa linha, RUESTCH (2005) apresenta um algoritmo intervalar híbrido (IA e busca local baseada no gradiente) que garante capturar todas as soluções de um MOP. A dificuldade do método consiste em reduzir o tamanho dos intervalos que contêm as soluções. O tamanho exagerado de um intervalo na verdade representa um pessimismo ou conservadorismo nos resultados.

De acordo com MOORE *et al.* (2009), quase todas computações científicas começam com valores inexatos para os dados e a IA fornece uma forma natural de incorporar as imprecisões de medição nos cálculos. Nesse contexto, SOARES *et al.* (2009c) propõem um método intervalar capaz de envolver a fronteira robusta de um RMOP. Na conclusão destacam que o sucesso do método depende da *qualidade das funções de inclusão*[2] para as funções de otimização e dos *parâmetros de precisão* do algoritmo. Além disso, ressaltam o alto esforço computacional despendido pelo método, principalmente, quando a dimensão do problema é alta ou necessita-se de um resultado bem acurado.

Pelo fato das técnicas puramente intervalares requererem alto tempo de processamento, o uso de técnicas híbridas é uma alternativa. Um algoritmo híbrido, por exemplo, um EA com a análise intervalar pode ser uma boa opção. Apesar de não se garantir a determinação do ótimo global, o EA-IA permite encontrar boas soluções em tempo razoável no tratamento de problemas com incertezas. Nessa linha, ROCCO e SALAZAR (2007) aplicaram um algoritmo híbrido (estratégias evolutivas e IA) em um problema de projeto robusto. A ideia foi usar a IA para determinar qual seria o máximo desvio que cada variável de decisão pode ter de modo que o projeto ainda atenda a determinados requisitos. Eles concluíram que o procedimento para checar a viabilidade do projeto é bom, pois

[2] Função intervalar associada a uma função real cuja imagem intervalar envolve (inclui) a imagem da função real. Na seção 5.3.3, ela é definida matematicamente.

utiliza apenas uma avaliação intervalar. O ponto negativo observado foi que o tamanho do intervalo de resposta para as variáveis pode ser superestimado. Apesar desse inconveniente, a expectativa é que o intervalo certamente conterá as soluções válidas.

Em SOARES (2008), foram propostos dois algoritmos para buscar soluções de RMOPs. Um deles o [I]RMOA II estritamente intervalar cujo objetivo é envelopar[3] as soluções robustas. O outro, denominado [I]RMOEA, é um algoritmo híbrido (evolucionário-intervalar) que objetiva encontrar as soluções robustas. Em termos de gasto computacional, o método híbrido é mais ágil do que o estritamente intervalar, além disso, segundo SOARES (2008), o [I]RMOEA é o mais promissor para tratar problemas multidimensionais. Nas conclusões é relatado que o processo de construção das funções de inclusão é uma das tarefas mais complicadas, principalmente, em problemas reais. A definição de solução robusta empregada em SOARES (2008) consiste na melhor solução para o pior caso de ação das incertezas. Sendo assim, tem-se que as soluções obtidas pelos métodos possuem o melhor desempenho possível para o pior caso das incertezas e são factíveis. Porém, pode acontecer de no pior caso de ação das incertezas a imagem da solução robusta estar fora da imagem viável.

Note que, como alertado antes, não há uma definição única de solução robusta. Prova disso é que a definição de (ROCCO e SALAZAR, 2007) difere da proposta por (SOARES, 2008). Pode-se perceber também que nos trabalhos descritos anteriormente os autores alcançam bons resultados e que a IA torna a computação segura frente às incertezas. Além disso, os trabalhos sugerem que um dos caminhos para avançar na área está relacionado com a construção e a qualidade das funções de inclusão. Ressalta-se que a obtenção da função de inclusão para cada uma das funções objetivo e restrições é um pré-requisito para que se possa resolver MOPs utilizando-se a IA. E, nem sempre, é fácil obter as funções de inclusão, principalmente, quando os valores das funções de otimização são gerados por meio de simulações computacionais.

Com isso, a estratégia de aliar EC e computação intervalar é uma alternativa promissora de avanço nas pesquisas para resolver um problema de relevância prática no mundo real que é a otimização robusta multiobjetivo. Entretanto, esta estratégia possui lacunas, principalmente, no que tange à obtenção e qualidade das funções de inclusão e a possibilidade da imagem da solução robusta estar fora da imagem viável. Desse modo, é relevante construir novos algoritmos evolucionários intervalares para RMOPs que utilizem técnicas capazes de lidar com essas dificuldades.

Diante do exposto, torna-se relevante investigar formas eficientes de utilizar EAs e IA para encontrar soluções robustas de bom desempenho em diversas situações pré-definidas de incerteza e que sejam factíveis, inclusive, no pior caso de atuação das incertezas.

[3] Envolver as soluções utilizando intervalos.

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver algoritmos evolucionários que utilizem análise intervalar de forma eficiente, para tratar incertezas paramétricas e ruído na solução de problemas de otimização robusta multiobjetivo.

1.2.2 Objetivos Específicos

Para atingir o objetivo geral, os seguintes objetivos específicos foram estabelecidos:

- fazer revisão bibliográfica dos assuntos otimização multiobjetivo, algoritmos evolucionários multiobjetivo, otimização robusta, análise intervalar e regressão simbólica;
- formular matematicamente o problema de otimização multiobjetivo robusta em termos das incertezas consideradas e da definição de solução robusta adotada;
- desenvolver algoritmos para aproximar funções de inclusão;
- validar os algoritmos desenvolvidos com problemas teste;
- aplicar os algoritmos propostos em problemas nas áreas de engenharia de controle e eletromagnetismo.

1.3 Delimitações

Algumas considerações quanto à delimitação do presente trabalho são as seguintes:

- por questões de padronização os problemas de otimização são apresentados no formato de minimização;
- a quantificação das incertezas é realizada por intervalos, não sendo considerada nenhuma probabilidade de ocorrência das incertezas;
- o termo intervalo refere-se a um intervalo conexo fechado;
- os métodos desenvolvidos baseados em IA trabalham com o conjunto dos intervalos reais. Intervalos complexos não são tratados;
- as funções de inclusão são obtidas pelo método da inclusão natural conforme explicado na Subseção 5.3.3;
- os algoritmos propostos foram implementados em Matlab;
- cada um dos assuntos da revisão bibliográfica é detalhado até o nível de profundidade suficiente para dar o suporte teórico necessário para as propostas deste trabalho. Referências adicionais estão disponibilizadas ao longo do texto para consultas mais detalhadas.

1.4 Contribuições

Na Tabela 1-1 são apontadas as principais contribuições desta tese. A primeira coluna contém a motivação para a contribuição, na segunda coluna tem-se um breve relato da contribuição, a terceira coluna indica a localização da descrição completa e da contribuição proposta.

Tabela 1-1- Principais contribuições desta tese.

Problema	Contribuição	Descrição
Algumas funções de otimização de MOPs não estão disponíveis analiticamente, sendo possível obter seus valores somente por meio de simulações. Ou então, a obtenção da função em uma forma analítica exija alto esforço matemático e tempo.	SNIF-GPA - Método mono-objetivo para encontrar funções de inclusão aproximadas. SNIF-MOGPA – Método multiobjetivo para encontrar funções de inclusão aproximadas.	Seções 7.1e 8.1
A função de inclusão disponível possui o intervalo de saída largo.	SNIF-MOGPA com um objetivo sendo minimizar a largura do intervalo de saída - Método multiobjetivo para aproximar funções de inclusão cuja largura do intervalo de saída seja mais próxima possível da largura mínima de referência.	Subseção 7.1.3 e Seção 8.3
Dependendo de como é definida a maximização das incertezas pode ocorrer o inconveniente da fronteira robusta gerada estar fora da imagem viável.	Definição da fronteira ideal de maximização.	Seções 7.2 e 8.4
Estender o uso de algoritmos evolucionários para problemas de otimização robusta multiobjetivo pela noção robusta *minimax*.	Algoritmo IRMOEA-M.	Seção 7.3, Subseção 8.2.3 e Seção 8.5
Estender o uso de algoritmos evolucionários para problemas de otimização robusta multiobjetivo pela noção robusta *minimax regret*.	Algoritmo IRMOEA-MR.	Seções 7.4 e 8.6

1.5 Organização do Trabalho

Os objetivos desta pesquisa serão alcançados com o estudo da teoria de otimização multiobjetivo, computação evolucionária, otimização robusta, análise intervalar e regressão simbólica. Inicialmente, no capítulo 2, define-se formalmente o problema de otimização multiobjetivo, os conceitos de dominância, solução Pareto ótima e outros itens relevantes. Em seguida, apresentam-se as formulações do RMOP considerando-se a noção robusta *minimax* e a *minimax regret*. No capítulo 3, realiza-se a revisão da bibliografia sobre incertezas e robustez, na qual apresentam-se os diferentes tipos de incertezas e formas de representá-las. Além disso, analisam-se as distintas abordagens de robustez e as formas mais utilizadas para tratá-las segundo a literatura. No capítulo 4, os conceitos fundamentais em EC são apresentados, enfatizando-se as principais características dos algoritmos genéticos. Em seguida, os principais algoritmos evolucionários multiobjetivo (*Multi-Objective Evolutionary Algorithms*- MOEAs) são descritos, bem como as características fundamentais que esses devem possuir. Ainda nesse capítulo, as principais formas de comparação de desempenho dos MOEAs são examinadas. Por último, expõe-se a técnica de programação genética. O capítulo 5

apresenta os principais conceitos envolvidos na análise intervalar. No capítulo 6, descrevem-se o problema de regressão simbólica e as principais técnicas empregadas na sua solução. Os algoritmos e conceitos propostos nesta tese são explicados no capítulo 7. O capítulo 8 traz os resultados, isto é, mostra a aplicação dos algoritmos propostos em problemas testes e reais. No capítulo 9, as considerações finais são feitas e as perspectivas de trabalhos futuros são apresentadas. Finalmente, as referências bibliográficas são descritas utilizando-se a norma ABNT NBR 6023:2002.

1.6 Conclusão

Neste capítulo mostrou-se a importância prática da consideração das incertezas nos problemas de otimização multiobjetivo. Além disso, as principais formas utilizadas de representação das incertezas foram citadas (intervalos, funções de pertinência e funções de densidade de probabilidade). Em seguida, a partir da representação escolhida para ser utilizada na presente pesquisa, descreveram-se as conclusões de alguns trabalhos relevantes encontrados na literatura explorando os pontos positivos e negativos encontrados. Com base nas dificuldades levantadas na utilização da análise intervalar, estabeleceram-se o objetivo geral e objetivos específicos desta pesquisa. Em sequência, apresentou-se o escopo, as contribuições e a organização do trabalho. Enfim, mostrou-se que a incorporação de incertezas em MOPs é relevante e que métodos híbridos envolvendo computação evolucionária e análise intervalar podem ser úteis, porém existem dificuldades a serem enfrentadas.

2. OTIMIZAÇÃO MULTIOBJETIVO

A otimização multiobjetivo é uma área da ciência que estuda, desenvolve e implementa algoritmos para a determinação do conjunto de soluções não-dominadas do problema analisado, segundo critérios pré-estabelecidos e que satisfaçam ao conjunto de restrições do modelo matemático. Esses critérios são conhecidos como funções objetivo, e caso sejam concorrentes, tornam o problema complexo, pois uma solução que melhora a avaliação de uma dada função objetivo gera a piora na avaliação de outra e vice-versa (*trade-off*). Segundo KNOWLES (2002) os problemas reais nas áreas de finanças, planejamento, engenharia, medicina, pesquisa operacional, entre outras, possuem objetivos múltiplos e conflitantes. Além disso, KNOWLES (2002) ressalta que, em geral, não há uma única solução capaz de otimizar todas as funções objetivo simultaneamente, mas sim um conjunto de soluções candidatas para as quais não se pode estabelecer um ordenamento total. Dessa forma, é necessária a atuação de um decisor para optar por uma das soluções candidatas de acordo com suas prioridades. Nesta tese, o foco principal é a busca por soluções robustas, sendo assim, a tomada de decisão não será considerada. Detalhes específicos sobre tomada de decisão podem ser encontrados em (PEDRYCZ *et al.*, 2011), (PARREIRAS e VASCONCELOS, 2009), (PARREIRAS, 2006), (COELLO *et al.*, 2007) e (MIETTINEN, 1999).

Este capítulo aborda a formulação matemática do MOP e os principais conceitos em otimização multiobjetivo.

A formulação do MOP conforme MIETTINEN (1999) é dada a seguir.

Definição 2-1 *O MOP, em termos de minimização, consiste em:*

$$\begin{aligned} &\textit{Minimizar } \boldsymbol{f}(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_{n_f}(\mathbf{x})\} \\ &\textit{Sujeito a } \mathbf{x} \in \mathbf{S}, \end{aligned} \tag{2-1}$$

sendo $\boldsymbol{f}(\mathbf{x})$ *o vetor de funções objetivo com* n_f (≥ 2) *elementos,* $\mathbf{x} = (\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_{n_v})^T$ *o vetor das variáveis de decisão e* $\mathbf{S} \subseteq \mathbb{R}^{n_v}$ *o conjunto viável.*

Para a definição do conjunto viável **S** é necessário estabelecer as funções de restrição do MOP (COELLO *et al.*, 2007).

As restrições de desigualdade do MOP são dadas por:

$$g_i(\mathbf{x}) \leq 0 \qquad i = 1, \dots, n_g. \tag{2-2}$$

E as restrições de igualdade são dadas por:

$$h_j(\mathbf{x}) = 0 \qquad j = 1, \dots, n_h. \tag{2-3}$$

Definição 2-2 *As restrições (2-2) e (2-3) definem sobre* $\mathbb{R}^{n_v}$ *o espaço viável* **S,**

$$\mathbf{S} = \{\mathbf{x} \in \mathbb{R}^{n_v} \mid \forall i \in \{1, \dots, n_g\} \wedge \forall j \in \{1, \dots, n_h\}, g_i(\mathbf{x}) \leq 0 \wedge h_j(\mathbf{x}) = 0\}. \tag{2-4}$$

Portanto, o espaço viável é a região do espaço $\mathbb{R}^{n_v}$ na qual todas as restrições do problema são satisfeitas. Nos problemas reais, as restrições são relacionadas às limitações de recursos disponíveis, limitações técnicas, características físicas do problema, entre outras.

Uma vez definido o conjunto viável, é possível definir a imagem viável.

Definição 2-3 *Dado* $\mathbf{z} = \boldsymbol{f}(\mathbf{x})$ *tem-se que a imagem viável* $\mathbf{Z} \subseteq \mathbb{R}^{n_f}$ *é dada por*

$$\mathbf{Z} = \boldsymbol{f}(\mathbf{S}) = \bigcup_{\mathbf{x} \in \mathbf{S}} \boldsymbol{f}(\mathbf{x}) \tag{2-5}$$

Em um MOP dois espaços Euclidianos são considerados: o espaço $\mathbb{R}^{n_v}$ das variáveis de decisão, no qual a coordenada em cada eixo corresponde a um componente do vetor **x**, e o espaço $\mathbb{R}^{n_f}$ dos objetivos, no qual a coordenada em cada eixo corresponde a um componente do vetor $\boldsymbol{f}(\mathbf{x})$. Conforme apresentado na Figura 2-1 cada ponto do espaço das variáveis de decisão representa uma solução e é relacionado a um ponto no espaço dos objetivos, no qual se avalia a qualidade da solução (COELLO *et al.*, 2007).

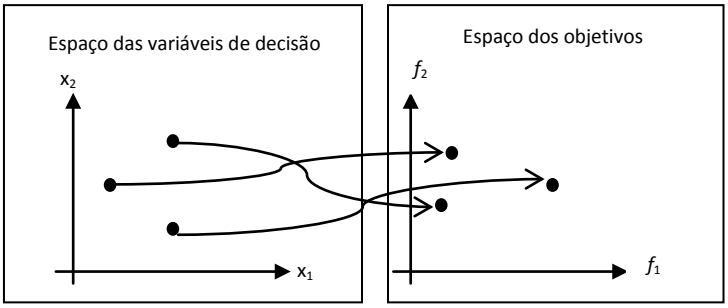


Figura 2-1 – Exemplo do espaço das variáveis de decisão (n=2) e do espaço dos objetivos (k=2).

Devido ao número de objetivos ser maior que 1 em um MOP, o processo de otimização ao invés de encontrar uma solução única, retorna um conjunto de soluções. Dessa forma, é necessário estabelecer a ordenação parcial das soluções obtidas considerando-se cada um dos objetivos. Para realizar tal ordenação é utilizado o conceito de dominância que segundo COELLO *et al.* (2007) é definido por:

Definição 2-4 *Um vetor* $\mathbf{u} = (\mathrm{u}_1, \mathrm{u}_2, \ldots, \mathrm{u}_{n_f})$ *domina outro vetor* $\mathbf{v} = (\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_{n_f})$, *representado por* $\mathbf{u} \prec \mathbf{v}$, *se e somente se,* $\forall i \in \{1, \ldots n_f\}, \mathrm{u_i} \leq \mathrm{v_i} \ \wedge \ \exists i \in \{1, .., n_f\} | \mathrm{u_i} < \mathrm{v_i}$.

Na Figura 2-2 verifica-se que todos os pontos da região 2 são dominados pelo ponto cinza e que todos os pontos da região 4 dominam o ponto cinza. Os pontos das regiões 1 e 3 não dominam e nem são dominados pelo ponto cinza. Percebe-se que para as soluções de um MOP não é possível estabelecer uma única solução que seja melhor que todas devido ao *trade-off* existente. O que conduz à existência de múltiplas soluções parcialmente ordenadas, as quais são não-dominadas entre si.

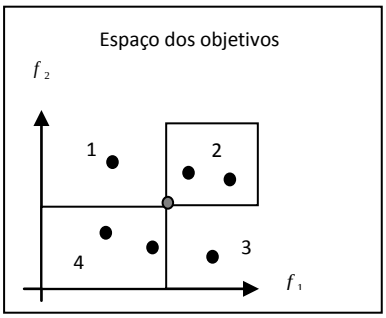


Figura 2-2 – Relações de dominância em relação ao ponto cinza.

A seguir, segundo MIETTINEN (1999) tem-se a definição de solução Pareto ótima que é um dos pilares da otimização multiobjetivo.

Definição 2-5 *Um vetor de decisão* $\mathbf{x}^* \in \mathbf{S}$ *é uma solução Pareto ótima se não existe outro vetor de decisão* $\mathbf{x} \in \mathbf{S}$ *tal que* $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)\ \forall i \in [1,2,..,n_f] \wedge \exists j \in [1,2,..,n_f] / f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ *para no mínimo um índice j.*

A Definição 2-5 expressa que o vetor $\mathbf{x}^*$ é Pareto ótimo se não existe nenhum outro vetor $\mathbf{x}$ no espaço viável que possa decrementar algum objetivo sem causar um simultâneo acréscimo em pelo menos outro objetivo. As soluções pertencentes ao conjunto Pareto ótimo são chamadas soluções Pareto ótimas, soluções não-inferiores, admissíveis ou eficientes (DIAS e VASCONCELOS, 2002).

Para um dado MOP cada uma das soluções Pareto ótimas quando avaliadas por $\boldsymbol{f}(\mathbf{x})$ gera vetores no espaço dos objetivos que em conjunto são conhecidos como fronteira Pareto ótima. Todos os vetores pertencentes à fronteira Pareto ótima são incomparáveis entre si. Na Figura 2-3, o espaço viável $\mathbf{S} \subset \mathbb{R}^2$ e sua imagem, a região viável no espaço dos objetivos $\mathbf{Z} \subset \mathbb{R}^2$, são exibidos. A parte de **Z** em negrito é a fronteira Pareto ótima.

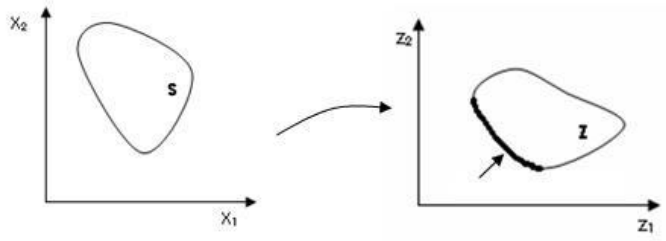


Figura 2-3 – Os conjuntos **S**, **Z** e a fronteira Pareto ótima.

2.1 O Problema de Otimização Robusta Multiobjetivo

Sejam $\boldsymbol{\delta}_\mathbf{x}$, $\boldsymbol{\delta}_\mathbf{e}$ e $\boldsymbol{\delta}_\mathbf{r}$ os vetores de incertezas, associados às variáveis de decisão, às condições do ambiente e ao ruído nas funções objetivo, respectivamente. Conforme sugerido em BEYER e SENDHOFF (2007), JIN e BRANKE (2005) e implementado por SOARES (2008), esses vetores podem ser unificados, uma vez que essas incertezas podem ser tratadas de forma semelhante. Sendo assim, define-se $\mathbf{P} \subseteq \mathbb{R}^{n_p}$ o conjunto de incertezas que engloba todos os vetores $\boldsymbol{\delta}_\mathbf{x}$, $\boldsymbol{\delta}_\mathbf{e}$ e $\boldsymbol{\delta}_\mathbf{r}$, sendo n_p a quantidade de parâmetros de incertezas. Sejam ainda dados o vetor de variáveis de decisão $\mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^{n_v}$, o vetor de incertezas $\mathbf{p} \in \mathbf{P} \subseteq \mathbb{R}^{n_p}$, o vetor de funções objetivo $\boldsymbol{f}(\mathbf{x}, \mathbf{p})$: $\mathbb{R}^{n_v} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_f}$, o vetor de funções de restrição de desigualdade $\boldsymbol{g}(\mathbf{x}, \mathbf{p})$: $\mathbb{R}^{n_v} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_g}$ e o vetor de funções de

restrição de igualdade $\boldsymbol{h}(\mathbf{x}, \mathbf{p})$: $\mathbb{R}^{n_v} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_h}$. Tendo em vista estas definições, apresentam-se as formulações do RMOP considerando-se duas variações da noção robusta de cenário de pior caso[4] utilizadas nesta tese, são elas: *minimax e minimax regret.*

2.1.1 Formulação Matemática do RMOP pela Noção Robusta do *minimax*

Um RMOP em termos de minimização pela noção robusta do *minimax* (robusto absoluto) é definido por:

$$\underset{\mathbf{x}\in \mathbf{S}}{\text{Min}} \ \underset{\mathbf{p}\in \mathbf{P}}{\text{Max}} \ \boldsymbol{f}(\mathbf{x}, \mathbf{p}) = \{f_1(\mathbf{x}, \mathbf{p}), \dots, f_{n_f}(\mathbf{x}, \mathbf{p})\} \tag{2-6}$$

sendo o espaço viável dado por:

$$\mathbf{S} = \{\, \mathbf{x} \in \mathbf{X} \,|\, \boldsymbol{g}(\mathbf{x}, \mathbf{p}) \leq \mathbf{0} \ \wedge \ \boldsymbol{h}(\mathbf{x}, \mathbf{p}) = \mathbf{0}, \forall\, \mathbf{p} \in \mathbf{P}\}. \tag{2-7}$$

A imagem viável sujeita à ação das incertezas é:

$$\mathbf{Z} = \ \boldsymbol{f}(\mathbf{S}, \mathbf{P}) = \bigcup_{\mathbf{x}\in \mathbf{S}, \mathbf{p}\in \boldsymbol{P}} \boldsymbol{f}(\mathbf{x}, \mathbf{p}). \tag{2-8}$$

Para resolver (2-6), deve-se encontrar o conjunto de minimizadores robustos para o pior caso da atuação das incertezas. Este conjunto é dado por

$$\mathbf{X}^* = \{\mathbf{x}^* \in \mathbf{S} \,|\, \nexists\, \mathbf{x} \in \mathbf{S}, \max_{\mathbf{p}\in \mathbf{P}} \boldsymbol{f}(\mathbf{x}, \mathbf{p}) \prec \max_{\mathbf{p}'\in \mathbf{P}} \boldsymbol{f}(\mathbf{x}^*, \mathbf{p}')\}. \tag{2-9}$$

2.1.2 Formulação Matemática do RMOP pela Noção Robusta do *minimax regret*

Um RMOP em termos de minimização pela noção robusta do *minimax regret* (desvio robusto) é estabelecido por:

$$\underset{\mathbf{x}\in \mathbf{S}}{\text{Min}} \ \underset{\mathbf{p}\in \mathbf{P}}{\text{Max}} \ \left(\boldsymbol{f}(\mathbf{x}, \mathbf{p}) \ominus \ \boldsymbol{f}^*(\mathbf{p})\right) \tag{2-10}$$

Com $\ominus$ sendo o operador de *regret*, definido em (2-13) e, $\boldsymbol{f}^*(\mathbf{p})$ representando o vetor que contém cada valor de referência (melhor valor) dos objetivos considerando-se o pior caso de atuação de cada incerteza $\mathbf{p}$ do conjunto $\mathbf{P}$, sendo:

$$\boldsymbol{f}^*(\mathbf{p}_\mathrm{j}) = \underset{\mathbf{x}\in \mathbf{S}}{\text{Min}} \ \underset{\mathbf{p}_\mathrm{j}}{\text{Max}} \ \ \boldsymbol{f}(\mathbf{x}, \mathbf{p}_\mathrm{j}), \tag{2-11}$$

com $\mathbf{p}_\mathrm{j}$ indicando a j-ésima incerteza do conjunto $\mathbf{P}$.

Assumindo que (2-11) retorne o conjunto de soluções não-dominadas $\mathbf{V}_\mathrm{j} = \{\mathbf{a}, \mathbf{b}, \dots, \mathbf{y}\} \subset \mathbb{R}^{n_f}$, o conjunto $\mathbf{V}_\mathrm{j}$ é reduzido a um ponto de referência $\mathbf{v} \in \mathbb{R}^{n_f}$ relativo à incerteza $\mathbf{p}_\mathrm{j}$ do conjunto $\mathbf{P}$ dado por

[4] É a abordagem robusta considerada no trabalho, explicada na seção 3.1.2.

$$\mathrm{v_k}(\mathbf{V_j}) = \min_{\mathrm{k}} \{a_k, b_k, \dots, y_k\}, k = 1, \dots, n_f \tag{2-12}$$

O operador de *regret* $\ominus$ ao ser aplicado em $\boldsymbol{f}(\mathbf{x}, \mathbf{p}) \ominus \boldsymbol{f}^*(\mathbf{p})$ resulta em:

$$\mathbf{r} = \begin{Bmatrix} max\left(f_1(\mathbf{x}, \mathbf{p}_1) - \mathrm{v}_1(\mathbf{V}_1), \dots, f_1(\mathbf{x}, \mathbf{p}_\mathrm{j}) - \mathrm{v}_1(\mathbf{V}_\mathrm{j}), \dots, f_1(\mathbf{x}, \mathbf{p}_{n_i}) - \mathrm{v}_1(\mathbf{V}_{n_i})\right), \\ \dots, \\ max\left(f_k(\mathbf{x}, \mathbf{p}_1) - \mathrm{v}_\mathrm{k}(\mathbf{V}_1), \dots, f_k(\mathbf{x}, \mathbf{p}_\mathrm{j}) - \mathrm{v}_\mathrm{k}(\mathbf{V}_\mathrm{j}), \dots, f_k(\mathbf{x}, \mathbf{p}_{n_i}) - \mathrm{v}_\mathrm{k}(\mathbf{V}_{n_i})\right), \\ \dots, \\ max\left(f_{n_f}(\mathbf{x}, \mathbf{p}_1) - \mathrm{v}_{n_f}(\mathbf{V}_1), \dots, f_{n_f}(\mathbf{x}, \mathbf{p}_\mathrm{j}) - \mathrm{v}_{n_f}(\mathbf{V}_\mathrm{j}), \dots, f_{n_f}(\mathbf{x}, \mathbf{p}_{n_i}) - \mathrm{v}_{n_f}(\mathbf{V}_{n_i})\right) \end{Bmatrix}, \tag{2-13}$$

com n_i indicando a quantidade de pontos de incerteza em **P.**

A imagem viável sujeita à ação das incertezas é:

$$\mathbf{Z} = \left(\boldsymbol{f}(\mathbf{S}, \mathbf{P}) \ominus \boldsymbol{f}^*(\mathbf{P})\right) = \bigcup_{\mathbf{x} \in \mathbf{S}, \mathbf{p} \in \boldsymbol{P}} \left(\boldsymbol{f}(\mathbf{x}, \mathbf{p}) \ominus \boldsymbol{f}^*(\mathbf{p})\right), \tag{2-14}$$

sendo **S** o espaço viável conforme definido em (2-7).

Para resolver (2-10), deve-se encontrar o conjunto de minimizadores robustos $\mathbf{X}^* \subseteq \mathbf{S}$ tal que

$$\mathbf{X}^* = \left\{\mathbf{x}^* \in \mathbf{S} \mid \nexists\, \mathbf{x} \in \mathbf{S}, \max_{\mathbf{p} \in \mathbf{P}}\left(\boldsymbol{f}(\mathbf{x}, \mathbf{p}) \ominus \boldsymbol{f}^*(\mathbf{p})\right) \prec \max_{\mathbf{p}' \in \mathbf{P}}\left(\boldsymbol{f}(\mathbf{x}^*, \mathbf{p}') \ominus \boldsymbol{f}^*(\mathbf{p}')\right)\right\}. \tag{2-15}$$

2.2 Conclusão

Neste capítulo descreveram-se as principais definições e conceitos que subsidiam a construção do modelo matemático do problema de otimização multiobjetivo. Em sequência, definiu-se a formulação do problema de otimização robusta multiobjetivo em termos das abordagens robustas *minimax* e *minimax regret* (descritas em mais detalhes na Subseção 3.1.2). Para a resolução de MOPs uma abordagem que vem sendo empregada na literatura é o uso de algoritmos não determinísticos como os métodos de computação evolucionária, que são apresentados no capítulo 4.

3. INCERTEZA E ROBUSTEZ

A pesquisa sobre os temas de incerteza e robustez traz consigo uma dificuldade inicial inerente, pois segundo VINCKE (2003) não existe uma única definição de robustez aceita pela comunidade científica. Sendo assim, é preciso inteirar-se das distintas definições presentes na literatura para verificar qual é a mais adequada ao contexto do problema que se deseja resolver. Caso nenhuma seja apropriada, deve-se então propor uma nova definição para lidar com o problema em mãos. Devido a essa característica, inicialmente será apresentado um breve histórico e em seguida alguns dos conceitos de robustez.

De acordo com MIETTINEN *et al.* (2008), a noção de robustez aplicada a problemas de decisão foi introduzida no trabalho de (GUPTA e ROSENHEAD, 1968). Conforme PERNY *et al*. (2006), o conceito de robustez foi introduzido em pesquisa operacional por (ROSENHEAD *et al.*, 1972) no contexto de planejamento dinâmico. BEYER e SENDHOFF (2007) afirmam que projetos de engenharia devem fornecer soluções com alto grau de robustez, sendo assim, tem-se a área de projeto robusto, a qual têm seus conceitos desenvolvidos de forma independente dos conceitos de robustez. Além disso, ainda alegam que essa área é altamente influenciada pela filosofia de projeto robusto de Taguchi (TAGUCHI, 1984). Certamente, o desenvolvimento em separado contribui para a existência de distinções conceituais.

No que tange aos conceitos relacionados a robustez, PERNY *et al*. (2006) dizem que o termo robusto tem sido usado para qualificar uma: *estratégia flexível*; *solução prudente*; *conclusão estável*. Dadas várias possíveis evoluções do contexto de decisão, a *estratégia flexível* preserva boas perspectivas. A *solução prudente* permanece satisfatória em todas as possíveis instâncias do problema. A *conclusão estável* permanece válida para múltiplas configurações realistas dos parâmetros de um modelo de decisão.

VINCKE (2003) aponta quatro conceitos de robustez: *decisão robusta*; *solução robusta*; *conclusão robusta*; *método robusto*. A *decisão robusta*, também chamada de flexível, é aquela que mantém abertos muitos bons planos para o futuro. A *solução robusta* é boa em todas ou na maioria das versões, sendo que versão é um conjunto plausível de valores para as incertezas do modelo. A *conclusão robusta* é a que é válida em todas ou na maioria das versões, sendo que versão é um conjunto aceitável para os parâmetros do modelo de decisão. O *método robusto* é o que retorna resultados válidos em todas ou na maioria das versões, uma versão é um conjunto dos valores possíveis para as incertezas do problema e para os parâmetros do método. VINCKE (2003) afirma que não há contradição entre os conceitos, pois esses apenas ilustram que diferentes tipos de robustez devem ser introduzidos no apoio à decisão. Além disso, para evitar confusão Vincke alerta para distinções entre robustez e análise de sensibilidade, pois a última envolve um estudo *a posteriori* realizado na vizinhança da solução de uma versão em particular. Enquanto a robustez considera diversas versões *a priori* e busca por soluções (decisões, conclusões) que são boas (válidas) em todas

ou na maioria das versões. Inclusive Vincke sugere evitar o uso da expressão *análise de robustez* uma vez que as considerações de robustez devem ser integradas durante o processo de apoio à decisão e não em uma análise posterior.

Nota-se que os conceitos relacionados à robustez apresentados até o momento avaliam a flexibilidade da *decisão robusta*, a qualidade e prudência da *solução robusta*, a validade e estabilidade da *conclusão robusta* e a validade do *método robusto*. No contexto de problemas de otimização, de acordo com VINCKE (2003) e PERNY *et al*. (2006), tem-se trabalhado com o conceito de *solução robusta*, ou seja, envolvendo qualidade e prudência. No entanto, conforme mostrado adiante, isso ainda não é garantia de uniformidade de conceitos, pois há na literatura distintos tipos de incertezas, bem como diferentes formas de quantificá-las.

BEYER e SENDHOFF (2007), no contexto de otimização de projeto robusto, apontam quatro tipos de incertezas:

(A) Mudança do ambiente e das condições de operação (envolvem a temperatura de operação, pressão, umidade, alterações nas propriedades dos materiais);

(B) Tolerâncias de produção e imprecisão do atuador (relacionada aos parâmetros de projeto do produto que só pode ser construído com certo grau de precisão. Como um maquinário de alta precisão é caro, um projeto menos sensível às tolerâncias de produção reduz custos. Esse tipo de incerteza é considerado uma perturbação nas variáveis de decisão);

(C) Incertezas na saída do sistema (relacionada às imprecisões no cálculo da saída do sistema, sejam por erros em medições ou de erros nas aproximações para os modelos originais);

(D) Incertezas de viabilidade (envolvem as restrições que as variáveis de projeto devem obedecer, geralmente aparecem junto com incertezas do tipo A e B).

Segundo BEYER e SENDHOFF (2007) existem três formas de quantificar as incertezas:

1) Tipo determinístico, aquele em que se define um domínio para os parâmetros dentro do qual as incertezas podem variar;
2) Tipo probabilístico, no qual medidas probabilísticas descrevem a probabilidade de certo evento ocorrer;
3) Tipo possibilístico, no qual são definidas medidas *fuzzy* que descrevem a possibilidade de certo evento acontecer. Dessa forma, afirmam que até doze (quatro tipo de incertezas multiplicado por três formas de quantificação) conceitos de robustez podem ser encontrados em problemas reais.

De acordo com BRANKE e JIN (2006), as incertezas são um fenômeno comum em problemas de otimização do mundo real, possuem muitas especificidades e aparecem de diversas maneiras. Segundo JIN e BRANKE (2005), as incertezas no contexto de otimização evolucionária podem ser

subdivididas em quatro classes: ruído, função *fitness* aproximada (metamodelo), função *fitness* variante no tempo (função *fitness* dinâmica) e robustez.

O ruído refere-se ao fato de que a função *fitness* está sujeita a perturbações provenientes de diversas fontes, como por exemplo, erros de sensores ou de simulações aleatórias. GOH e TAN (2009) ressaltam que na situação de ruído, a incerteza é inerente às funções objetivo e tende a conduzir o processo de otimização a erros. Matematicamente uma função *fitness* com ruído ($\tilde{f}_r(\mathbf{x})$) é descrita em (JIN e BRANKE, 2005) como:

$$\tilde{f}_r(\mathbf{x}) = \int_{-\infty}^{+\infty} [\tilde{f}(\mathbf{x}) + \delta_r] p(\delta_r) d\delta_r, \qquad \delta_r \sim N(0, \sigma^2), \tag{3-1}$$

sendo $\mathbf{x}$ o vetor de variáveis de decisão, $\tilde{f}(\mathbf{x})$ a função *fitness*, δ_r o ruído, o qual geralmente é assumido ser normalmente distribuído com média zero e variância σ^2 e $p(\delta_r)$ a distribuição de probabilidade de δ_r.

A função *fitness* aproximada (metamodelo) consiste em aproximar a função *fitness* por meio de simulações ou de dados gerados empiricamente. É útil nos casos em que não é possível obter a função *fitness* analiticamente ou é computacionalmente caro avaliar a função *fitness* original. Sendo assim, a função *fitness* aproximada ($\tilde{f}_a(\mathbf{x})$) a ser otimizada pelos algoritmos evolucionários seria (JIN e BRANKE, 2005):

$$\begin{aligned} &\tilde{f}_a(\mathbf{x}) = \tilde{f}(\mathbf{x}), \text{ se a função } \textit{fitness} \text{ original é usada} \\ &\tilde{f}_a(\mathbf{x}) = \tilde{f}(\mathbf{x}) + E(\mathbf{x}), \text{ se um metamodelo é usado} \end{aligned} \tag{3-2}$$

sendo $\tilde{f}(\mathbf{x})$ a função *fitness* original e $E(\mathbf{x})$ o erro de aproximação do metamodelo. A principal diferença entre $\tilde{f}_r(\mathbf{x})$ e $\tilde{f}_a(\mathbf{x})$ é que o erro na segunda é determinístico (uma vez que o metamodelo foi idealizado) e regular (a média do erro não é zero). Com isso, não se pode reduzir o erro recalculando-se $\tilde{f}_a(\mathbf{x})$. O erro deve ser tratado utilizando-se a função *fitness* original. O desafio nessa classe é encontrar um equilíbrio entre avaliações com a função original (alta qualidade e custo elevado) e as avaliações com a função aproximada (erro inerente e baixo custo).

Na classe de função *fitness dinâmica* tem-se que a função objetivo é determinística em qualquer ponto do tempo, no entanto, é descrita em função do tempo t. A função *fitness dinâmica* ($\tilde{f}_d(\mathbf{x})$) é definida por (JIN e BRANKE, 2005):

$$\tilde{f}_d(\mathbf{x}) = \tilde{f}_t(\mathbf{x}). \tag{3-3}$$

Com isso as soluções ótimas também mudam com o tempo. Nesse caso, o algoritmo deve ser capaz de acompanhar a solução ótima ao invés de ser reexecutado a cada instante. Sendo assim, é essencial desenvolver mecanismos para reutilizar a informação de ambientes anteriores a fim de tornar o processo de otimização ágil após alguma mudança.

Considerando-se a robustez, tem-se que as variáveis de projeto e/ou parâmetros do ambiente (condições operacionais) estão sujeitos a perturbações *depois* que a solução ótima foi determinada (JIN e BRANKE, 2005). Dessa forma, um requisito comum é que a solução deve funcionar satisfatoriamente quando as variáveis de projeto e/ou parâmetros do ambiente mudarem levemente de valor. Tais soluções são chamadas de soluções robustas. De acordo com JIN e BRANKE (2005), algoritmos evolucionários para buscar soluções robustas devem trabalhar com a seguinte função *fitness*:

$$\tilde{f}_{xe}(\mathbf{x}) = \int_{-\infty}^{+\infty} [\tilde{f}(\mathbf{x} + \boldsymbol{\delta}_{\mathbf{xe}})] p(\boldsymbol{\delta}_{\mathbf{xe}}) d\boldsymbol{\delta}_{\mathbf{xe}}, \tag{3-4}$$

sendo $\boldsymbol{\delta}_{\mathbf{xe}}$ o vetor de perturbações relacionado às incertezas das variáveis de projeto e/ou dos parâmetros do ambiente e $p(\boldsymbol{\delta}_{\mathbf{xe}})$ a distribuição de probabilidade das perturbações. Note que, se as perturbações são analisadas *a posteriori*, tal procedimento assemelha-se à análise de sensibilidade, conforme alertado por (VINCKE, 2003).

Comparando as abordagens de BEYER e SENDHOFF (2007) e de JIN e BRANKE (2005) nota-se o seguinte:

I. Para JIN e BRANKE (2005), as soluções robustas são aquelas relacionadas às perturbações nas variáveis de projeto e/ou parâmetros do ambiente. Diferentemente, BEYER e SENDHOFF (2007) consideram a robustez em todos os tipos de incerteza (A, B, C e D), as quais podem ser tomadas em conjunto ou em separado. Percebe-se que se forem consideradas as incertezas do tipo A e B com a quantificação dessas pelo método probabilístico, os conceitos em ambos os trabalhos seriam semelhantes. Também se pode verificar que a incerteza do tipo C engloba tanto as incertezas relacionadas ao ruído quanto ao erro referente à função *fitness* aproximada. Portanto, embora o contexto dos trabalhos seja diferente, sabendo juntar as peças nota-se a proximidade das ideias. Jin e Branke deram pistas disso, uma vez que nos tópicos de pesquisa futura do seu trabalho sugerem lidar com mais de um aspecto de incerteza simultaneamente. Inclusive, fornecem como exemplo a busca por soluções robustas em otimização de projetos.
II. BEYER e SENDHOFF (2007) não consideram a otimização dinâmica, provavelmente deve-se ao fato desse tipo de incerteza não fazer sentido no contexto de projeto robusto.
III. JIN e BRANKE (2005) focaram estritamente no tratamento probabilístico das incertezas.
IV. JIN e BRANKE (2005) não consideram explicitamente a incerteza do tipo D. Não considerar esse tipo de incerteza é admitir a possibilidade de uma eventual perturbação na solução torná-la inviável, o que em geral, é indesejável. Ressalta-se que em (AVIGAD e BRANKE, 2008) a incerteza do tipo D é considerada.

Na próxima seção, a otimização robusta, que é o escopo desta tese, será tratada com mais profundidade.

3.1 Otimização Robusta

Como visto anteriormente, na literatura são apresentadas diversas noções de robustez, o que gera distintos conceitos para as soluções robustas. A seguir, são apresentados alguns desses conceitos:

- No contexto de projeto robusto, soluções robustas são aquelas cujos valores dos parâmetros de projeto bem como o desempenho do projeto se mantêm relativamente inalterados quando expostos às condições de incerteza (BEYER e SENDHOFF, 2007).
- A busca por soluções robustas visa assegurar que os requisitos de desempenho são encontrados e que as restrições não são violadas devido às incertezas do sistema (AVIGAD e BRANKE, 2008).
- Um projeto robusto é aquele no qual o desempenho se mantém relativamente inalterado e viável na presença de incertezas (MATTSON e MESSAC, 2005).
- Soluções robustas são aquelas menos sensíveis a pequenas perturbações nas variáveis dentro de sua vizinhança (DEB e GUPTA, 2006).
- Projeto robusto implica na robustez do objetivo e das funções de restrição. A robustez da função objetivo faz com que o desempenho do sistema fique insensível às variações nas variáveis de projeto. A robustez da função de restrição indica que considerando as tolerâncias o ótimo é viável (LEE e PARK, 2001).
- Projetos robustos não violam as especificações e não alteram muito o seu desempenho na presença de incertezas (ONG *et al.*, 2006).

Nesta tese, decidiu-se por desenvolver métodos construídos especificamente para otimização multiobjetivo robusta, segundo os conceitos *minimax* e *minimax regret*, explicados a seguir na Subseção 3.1.2, sendo a computação de incertezas realizada utilizando-se de técnicas intervalares.

Dadas as restrições e os limites das variáveis de decisão, é possível verificar a factibilidade de uma dada solução. Para que seja possível avaliar uma solução em termos de desempenho é necessário definir como esse será mensurado. É nesse ponto que entram as medidas robustas. GOH e TAN (2009) chamam de medidas robustas as noções de robustez que são aplicadas no contexto de otimização robusta.

Conforme GOH e TAN (2009) as medidas robustas mais comuns são a função *fitness* esperada[5] (JIN e BRANKE, 2005; PAENKE *et. al*, 2006) e o cenário de pior caso (AVIGAD e BRANKE, 2008; SOARES *et al.*, 2009a), os quais serão abordadas em seções específicas adiante. Há outras diversas

[5] Tradução do termo em inglês *expected fitness function* utilizado na literatura.

medidas robustas encontradas na literatura, por exemplo, soluções robustas do tipo 1 (que é baseada na função *fitness* esperada) e soluções robustas do tipo 2 (que consiste em adicionar ao modelo matemático do problema uma restrição relacionada à robustez) definidas em (DEB e GUPTA, 2006), abordagens multiobjetivo que resultam em *trade-offs* entre desempenho e robustez (LEE e PARK, 2001). Nesse último caso, a robustez pode ser medida como uma combinação da média e do desvio-padrão da função objetivo ou como a variância da *fitness*.

Outro ponto relevante em otimização robusta é a forma de quantificar as incertezas. Nesse aspecto, percebe-se na literatura que o tipo probabilístico e o tipo determinístico são mais utilizados em relação ao tipo possibilístico em abordagens evolucionárias multiobjetivo.

3.1.1 Otimização Robusta Baseada na Função *fitness* Esperada

A otimização robusta baseada na função *fitness* esperada considera as incertezas como números aleatórios que seguem uma dada distribuição de probabilidade, a qual reflete o conhecimento real sobre as mesmas (BEYER e SENDHOFF, 2007).

Conforme JIN e BRANKE (2005), nesta medida robusta cada função *fitness* original é substituída por sua respectiva função *fitness* esperada, também chamada de função *fitness* efetiva[6], que no caso de incertezas relacionadas às variáveis de decisão e/ou parâmetros do ambiente é dada pela Equação (3-4), a qual é repetida a seguir por conveniência.

$$\tilde{f}_{xe}(\mathbf{x}) = \int_{-\infty}^{+\infty} [\tilde{f}(\mathbf{x} + \boldsymbol{\delta}_{\mathbf{xe}})] p(\boldsymbol{\delta}_{\mathbf{xe}}) d\boldsymbol{\delta}_{\mathbf{xe}}, \tag{3-4}$$

Geralmente, a função *fitness* efetiva é estimada. Isso se deve ao fato de que para problemas relativamente complexos esta função, em geral, não pode ser encontrada em uma forma analítica fechada. Para fazer essa estimativa, em geral, utiliza-se da técnica de média explícita ou da média implícita (JIN e BRANKE, 2005), as quais são descritas a seguir.

3.1.1.1 Média Explícita

Na técnica de média explícita, o modo mais simples é fazer a estimativa pela integração de Monte Carlo, isto é, por meio da média dos valores da função *fitness* sobre um número de amostras de perturbações geradas aleatoriamente na vizinhança da solução a ser avaliada. Considerando-se a incerteza nas variáveis de decisão e/ou parâmetros do ambiente, a função *fitness* efetiva pela técnica de Monte Carlo é:

$$\hat{f}(\mathbf{x}) = \frac{1}{n} \sum_{j=1}^{n} \tilde{f}\left(\mathbf{x} + \boldsymbol{\delta}_{\mathbf{xe_j}}\right), \tag{3-5}$$

[6] Tradução do termo em inglês *effective fitness function* utilizado na literatura.

sendo $\hat{f}(\mathbf{x})$ a função *fitness* efetiva estimada para a função *fitness* $\tilde{f}$, n a quantidade de amostras e $\boldsymbol{\delta}_{\mathbf{xe_j}}$ o j-ésimo vetor de perturbação nas variáveis de decisão e/ou parâmetros do ambiente.

Apesar da fácil implementação, na integração de Monte Carlo a quantidade de amostras influi na precisão da estimativa e no custo computacional. Um menor n implica em uma baixa qualidade na estimativa devido à existência de alta variância. Há estudos sobre formas de reduzir o número de amostras, consequentemente o esforço computacional e, mesmo assim, tentar obter estimativas com boa precisão. Além disso, na técnica de Monte Carlo cada amostra corresponde a uma avaliação da função *fitness* original. Caso o cálculo da função *fitness* original seja dispendioso em termos computacionais, essa técnica é claramente desvantajosa. Nesse caso, recomenda-se usar funções *fitness* aproximadas ao invés de utilizar uma simulação computacionalmente cara. A função aproximada pode ser obtida a partir de dados que mapeiem as variáveis de decisão no valor de *fitness*. Quando as aproximações são obtidas por meio desse mapeamento dos dados, são chamadas de metamodelos ou *surrogates*. Há diversos métodos utilizados para esse fim: polinomiais (conhecidos como *response surface methodologies*), *perceptrons* multi-camadas, redes neurais do tipo *radial-basis-fuction*, *kriging*, programação genética, entre outros. Em metamodelos, em geral, obtêm-se aproximações que representam um *trade-off* entre qualidade e complexidade (JIN e BRANKE, 2005). Uma boa revisão sobre construção de *surrogates* pode ser vista em (GORISSEN, 2010).

Há outras técnicas baseadas em *fitness* aproximadas específicas para EAs que atuam do seguinte modo: para estimar a *fitness* esperada de um indivíduo usam-se avaliações de indivíduos similares calculadas anteriormente. Essas técnicas no contexto dos EAs são conhecidas como *fitness inheritance*, *fitness imitation* e *fitness assignment* (JIN e BRANKE, 2005). Uma técnica que se baseia nessa ideia é proposta em (BRANKE, 1998).

3.1.1.2 Média Implícita

A técnica de média implícita consiste em definir um tamanho grande para a população nos algoritmos genéticos, usar seleção proporcional (por exemplo, a roleta) e, a cada geração, adicionar perturbações aleatórias às variáveis de decisão. A base dessa técnica é que um alto número de indivíduos é capaz de reduzir a influência das incertezas e assegurar a convergência do algoritmo (JIN e BRANKE, 2005). Existem trabalhos que mostram que a média implícita tem o mesmo efeito que otimizar pela função *fitness* efetiva (TSUTSUI *et al.*, 1996; TSUTSUI e GHOSH, 1997).

De acordo com RUSTEM e HOWE (2002), o principal problema da abordagem do desempenho esperado é que essa negligencia a ocorrência do cenário de pior caso do efeito das incertezas. Com isso, uma solução robusta baseada na função *fitness* efetiva possui chances de conduzir o sistema otimizado ao fracasso, podendo resultar em consequências drásticas ao sistema.

3.1.2 Otimização Robusta Baseada no Cenário de Pior Caso

Segundo GOH *et al.* (2007) essa abordagem visa garantir a confiabilidade máxima, isso quer dizer, assegurar que, mesmo para o pior caso possível da atuação das incertezas, as soluções robustas encontradas terão sucesso total na resolução do problema considerado. Diante disso, o uso dessa abordagem é recomendado caso o problema exija soluções seguramente imunes às incertezas.

Conforme proposto por KOUVELIS e YU (1997) deve-se identificar os potenciais cenários do problema e em seguida se preocupar em obter a melhor solução associada especificamente ao pior cenário identificado. Para obtenção do pior cenário, KOUVELIS e YU (1997) definem três critérios: *robusto absoluto*, *desvio robusto* e *robusto relativo*. O primeiro utiliza o *minimax* e os dois últimos o *minimax regret*. No critério *robusto absoluto*, a solução robusta é aquela cujo maior (pior) valor da função objetivo considerando-se todos os cenários (ou seja, o valor do objetivo no pior cenário) é o menor possível. No *minimax regret*, o primeiro passo é computar o *regret* (arrependimento) de cada decisão em cada cenário. Há duas variações (*desvio robusto* e *robusto relativo*) dependendo de como se calcula o *regret*. No *desvio robusto*, o arrependimento é definido como a diferença entre o valor da função objetivo de uma decisão específica e o valor da função objetivo da decisão ótima para um dado cenário. No *robusto relativo*, o arrependimento é calculado pela razão entre o valor da função objetivo de uma decisão específica e o valor da função objetivo da decisão ótima para um dado cenário. Tanto no *desvio robusto* quanto no *robusto relativo*, depois que o *regret* é obtido, tem-se que aplicar o *minimax*, ou seja, escolher a decisão com o menor máximo *regret*. O que esses critérios indicam é que, diante de incertezas, é necessário considerar todas as situações possíveis, inclusive a pior, uma vez que não se sabe qual delas poderá acontecer.

Para apresentar a formulação matemática dos critérios propostos em (KOUVELIS e YU, 1997) faz-se necessário estabelecer que $\mathbf{C}$ é o conjunto de todos os cenários possíveis, $\mathbf{X}$ é o conjunto de variáveis de decisão, $\mathbf{D}$ é o conjunto de incerteza nos dados de entrada, $\mathrm{d_c}$ é a incerteza associada ao cenário c e $\mathbf{F_c}$ representa o conjunto de todas as soluções factíveis quando o cenário c ocorre. Além disso, suponha que a qualidade da solução $\mathrm{x} \in \mathbf{F_c}$ é avaliada usando a função $f(\mathrm{x}, \mathrm{d_c})$. Então a solução ótima $\mathrm{x_c^*}$ para a incerteza $\mathrm{d_c}$ satisfaz

$$\mathrm{z_c} = f(\mathrm{x_c^*}, \mathrm{d_c}) = \min_{\mathrm{x} \in \mathbf{F_c}} f(\mathrm{x}, \mathrm{d_c}) \tag{3-6}$$

A solução pelo critério *robusto absoluto* $\mathrm{x_a}$ é a que minimiza o valor de f, entre todas as soluções viáveis considerando todos os cenários de incerteza, isto é,

$$\mathrm{z_a} = \max_{\mathrm{c} \in \mathbf{C}} f(\mathrm{x_a}, \mathrm{d_c}) = \min_{\mathrm{x} \in \mathbf{F_c}} \max_{\mathrm{c} \in \mathbf{C}} f(\mathrm{x}, \mathrm{d_c}) \tag{3-7}$$

A solução utilizando-se o critério *desvio robusto* x_d é a que possui o menor desvio no pior caso em relação à solução ótima, entre todas as soluções viáveis considerando todos os cenários de incerteza, isto é,

$$z_d = \max_{c \in \mathbf{C}} (f(x_d, d_c) - f(x_c^*, d_c)) = \min_{x \in \mathbf{F}_c} \max_{c \in \mathbf{C}} (f(x, d_c) - f(x_c^*, d_c)) \tag{3-8}$$

A solução utilizando-se o critério *robusto relativo* x_r é a que possui o menor desvio relativo no pior caso em relação à solução ótima, entre todas as soluções viáveis considerando todos os cenários de incerteza, isto é,

$$z_r = \max_{c \in \mathbf{C}} \left(\frac{f(x_r, d_c) - f(x_c^*, d_c)}{f(x_c^*, d_c)} \right) = \min_{x \in \mathbf{F}_c} \max_{c \in \mathbf{C}} \left(\frac{f(x, d_c) - f(x_c^*, d_c)}{f(x_c^*, d_c)} \right) \tag{3-9}$$

Em (3-9) tem-se que $f(x_c^*, d_c)$ deve ser diferente de zero.

As soluções obtidas com o *desvio robusto* e o *robusto relativo* são menos conservadoras do que as obtidas com o *robusto absoluto*, uma vez que levam em conta o valor da solução ótima em cada um dos possíveis cenários. O *robusto absoluto* visa obter a melhor solução para o pior cenário de atuação das incertezas.

Para RUSTEM e HOWE (2002), o cenário de pior caso é importante nas decisões que podem acarretar consequências catastróficas ao sistema otimizado, contudo alertam que podem existir situações nas quais o pior caso seja tão improvável de ocorrer, que o uso dessa medida robusta poderá resultar em decisões pessimistas desnecessárias.

De acordo com AVIGAD e BRANKE (2008) o uso do cenário de pior caso dentro do processo de busca em otimização evolucionária multiobjetivo é dificilmente encontrado. Nos parágrafos seguintes são apresentadas algumas das abordagens existentes na literatura.

No trabalho de (GOH e TAN, 2007) considera-se que as incertezas estão associadas às variáveis de decisão e/ou parâmetros do ambiente e é proposto um algoritmo evolucionário multiobjetivo robusto denominado RMOEA para encontrar *tradeoffs* entre Pareto-otimalidade e robustez. Nesse caso, a noção de robustez utilizada é a do cenário de pior caso, e o seu cálculo consiste em determinar para cada objetivo qual a variação percentual do pior caso em relação ao valor da solução não perturbada. Note que essa definição parece o critério *robusto relativo*, no entanto, difere pelo fato de que o critério *robusto relativo* calcula a variação relativa do pior caso em relação à solução ótima do cenário. Para que a robustez seja considerada, acrescenta-se ao problema original um objetivo adicional que lida somente com a robustez. O valor desse objetivo adicional é definido pela pior variação dentre as variações obtidas para cada um dos objetivos do problema original. Para determinar o cenário de pior caso aplica-se dentro do RMOEA o algoritmo µGA (*Micro Genetic Algorithm*), o

qual desenvolve uma busca local em uma vizinhança pré-estabelecida para obter o desempenho no pior caso de incerteza de cada indivíduo da população.

Em (AVIGAD e BRANKE, 2008), os autores focam na busca por soluções robustas cujas perturbações ocorrem tanto nas variáveis de projeto quanto nos parâmetros ambientais. É assumido que os valores das perturbações estão dentro de uma faixa conhecida de antemão. A definição de robustez utilizada é o critério *robusto absoluto* (*minimax*). O método de resolução proposto consiste de um algoritmo evolucionário multiobjetivo principal para realizar a parte de minimização do *minimax*. A parte relativa à maximização das incertezas é desenvolvida por meio de um algoritmo evolucionário que é acoplado ao algoritmo evolucionário principal. É assumido nessa etapa que o pior caso não pode ser resumido a uma única solução, portanto tem-se uma fronteira com os piores casos de cada solução. Nessa abordagem a maior desvantagem é que o algoritmo evolucionário acoplado tem que ser executado para cada indivíduo da população do algoritmo principal, o que é computacionalmente caro.

Tanto o método proposto em (GOH e TAN, 2007), quanto em (AVIGAD e BRANKE, 2008), a ideia principal é baseada no uso de algoritmos evolucionários para se calcular o pior caso.

SOARES *et al.* (2009a) propõem um algoritmo genético multiobjetivo para lidar com incertezas relacionadas às variáveis de decisão em um problema de eletromagnetismo. A definição de robustez utilizada é o critério *robusto absoluto* (*minimax*). A abordagem proposta consiste em aplicar um algoritmo genético multiobjetivo, sendo que no cômputo de uma aproximação para o pior caso, cada indivíduo sofre um número finito de perturbações aleatórias. Com isso, pode-se determinar o chamado ponto de pior caso, o qual funciona como um ponto ideal de maximização e serve como uma referência para guiar a busca. Nessa abordagem verifica-se que a qualidade da aproximação depende da quantidade de perturbações realizadas.

Em (SOARES, 2008), é apresentada a solução de um problema de sintonia de controlador PD (Proporcional-Derivativo), no qual existem imprecisões nos parâmetros de ganho do controlador. A definição de robustez utilizada é o critério *robusto absoluto* (*minimax*). Os métodos aplicados no problema são denominados [I]RMOA I, [I]RMOA II e [I]RMOEA. Os três algoritmos utilizam análise intervalar para computar as incertezas e fazem uso do ponto ideal de maximização para determinar o pior caso de cada solução. O primeiro algoritmo, a partir de uma precisão fornecida, define um envelope sobre a fronteira robusta. O segundo algoritmo envelopa todas as soluções robustas. O terceiro é um algoritmo híbrido que associa técnicas intervalares aos algoritmos genéticos para encontrar as soluções robustas. Os algoritmos funcionaram bem, sendo capazes de gerar respostas eficientes para o problema. Uma dificuldade da abordagem foi a de encontrar as funções objetivo e restrições do problema em uma forma analítica, o que é necessário para o emprego das técnicas intervalares. Uma vantagem da abordagem é que o tratamento das incertezas por meio da análise intervalar, realmente permite computar o pior caso de atuação das incertezas de acordo com as precisões definidas no algoritmo. É importante ressaltar que o uso do ponto ideal de maximização

para determinar o pior caso de atuação das incertezas das soluções pode levar a situações em que a solução robusta não pertence ao espaço viável. Diante disso, é importante pesquisar outras formas de estabelecer o pior caso de ação das incertezas.

Para o desenvolvimento deste capítulo diversos trabalhos relevantes foram consultados na literatura. A Tabela 3-1 sumariza as principais publicações em termos: da característica do trabalho (se é uma aplicação ou um *survey*); da natureza do problema abordado (mono ou multiobjetivo); da medida de robustez empregada (por exemplo, *fitness* efetiva, pior caso, abordagem multiobjetivo, restrição adicional); e da metodologia de resolução proposta (MOEA, entre outras). Percebe-se que a maioria das publicações está relacionada a aplicações em problemas multiobjetivo que utilizam MOEAs na resolução. Quanto à medida de robustez a maior parte diz respeito a *fitness* efetiva e aproximação do pior caso. Ressalva-se o seguinte, nos trabalhos associados à aproximação do pior caso, com exceção de (SOARES *et al.*, 2009a), todos os autores intitulam o trabalho original de pior caso, sem o termo aproximação. No entanto, na Tabela 3-1 o termo aproximação é utilizado, pois as metodologias empregadas nesses trabalhos não asseguram que realmente atinge-se o pior caso. Na verdade, têm-se boas estimativas. Destaca-se também que a abordagem de pior caso utilizada em todos os trabalhos listados é o *minimax* (*robusto absoluto*).

Tabela 3-1- Perfil dos principais trabalhos consultados na revisão sobre otimizacão robusta.

Publicação	**Característica**	**Natureza do Problema**	**Medida de Robustez**	**Metodologia Abordada**
(AVIGAD e BRANKE, 2008)	Aplicação	Multi	Aproximação do pior caso	MOEA com NSGA-II embutido
(BARRICO e ANTUNES, 2007)	Aplicação	Multi	*Fitness* efetiva	MOEA
(BEYER e SENDHOFF, 2007)	*Survey*	Mono e Multi	*Minimax*, medidas de probabilidade (*fitness* efetiva), formulação multiobjetivo, restrição probabilística a ser satisfeita	Programação matemática, EA, MOEA
(DEB e GUPTA, 2006)	Aplicação	Multi	*Média da fitness* efetiva e restrição adicional	MOEA
(GOH e TAN, 2007)	Aplicação	Multi	Aproximação do pior caso e formulação multiobjetivo	MOEA com µGA embutido
(GOH *et al.*, 2007)	Aplicação	Multi	*Fitness* efetiva	MOEA
(GUIMARÃES *et al.*, 2006a)	Aplicação	Multi	Formulação multiobjetivo	MOEA
(HO *et al.*, 2008)	Aplicação	Mono	*Fitness* efetiva	Busca Tabu
(HO e YANG, 2010)	Aplicação	Mono	*Fitness* efetiva para a função objetivo e pior caso para as restrições	EA com aprendizado competitivo
(JIN e BRANKE, 2005)	*Survey*	Mono e Multi	*Fitness* efetiva e formulação multiobjetivo	EA e MOEA
(LEE e PARK, 2001)	Aplicação	Multi (ponderado)	Formulação multiobjetivo	Programação matemática
(MENDES *et al.*, 2012c)	Aplicação	Multi	Aproximação do pior caso	MOEA
(ONG *et al.*, 2006)	Aplicação	Mono	Aproximação do Pior Caso	EA
(PAENKE *et al.*, 2006)	Aplicacão	Mono e Multi	*Média da fitness* efetiva e formulação multiobjetivo	EA e MOEA

(PFLUGFELDER *et al.*, 2008)	Aplicação	Mono	Aproximação do Pior Caso	L-BFGS-B[7]
(ROCCO e SALAZAR, 2007)	Aplicação	Multi	Desvio máximo de cada variável de modo que o projeto atenda as especificações	MOEA com IA
(SOARES, 2008)	Aplicação	Multi	Pior caso	Somente IA e MOEA com IA
(SOARES *et al.*, 2009a)	Aplicação	Multi	Aproximação do pior caso	MOEA
(SOARES *et al.*, 2009b)	Aplicação	Multi	Pior caso	MOEA com IA
(SOARES *et al.*, 2009c)	Aplicação	Multi	Pior caso	Somente IA
(TSUTSUI e GHOSH, 1997)	Aplicação	Mono	*Fitness* efetiva	EA
(ZANG *et al.*, 2002)	*Survey*	Multi (ponderado)	Formulação multiobjetivo	*Sampling methods*

3.2 Conclusão

Neste capítulo diversas noções de robustez encontradas na literatura foram apresentadas. As diferentes formas de incertezas relatadas nas publicações examinadas foram discutidas. Além disso, os principais métodos empregados para lidar com o problema de otimização robusta foram considerados. Por fim, as principais características de um conjunto de publicações relevantes na área foram sucintamente resumidas e apresentadas em uma tabela. Percebe-se que, apesar das primeiras noções de robustez terem surgido há cerca de 40 anos atrás, a preocupação de incorporá-las em problemas de otimização é um assunto recente, visto que a maioria das publicações é atual. Salienta-se também, que devido aos inúmeros conceitos de robustez, podem ser obtidas soluções robustas distintas para um mesmo RMOP, afinal o resultado depende da medida robusta considerada. Além disso, as medidas robustas têm comportamentos diferentes, algumas são mais conservadoras, outras nem tanto. A questão chave é que do ponto vista prático, em ambientes incertos, alguma medida robusta adequada à natureza do problema tratado deve ser adotada.

[7] O L-BFGS-B é uma variante do algoritmo *Limited Memory* Broyden–Fletcher–Goldfarb–Shanno (L-BFGS). Essa variante é capaz de lidar com restrições nas quais as variáveis têm limite inferior e superior, mais detalhes em (BYRD *et al.*, 1995).

4. COMPUTAÇÃO EVOLUCIONÁRIA

Neste capítulo, inicialmente, apresentam-se os fundamentos da EC, em seguida, apontam-se referências bibliográficas para consultas detalhadas sobre a mais conhecida das técnicas evolucionárias (algoritmos genéticos), sua estrutura de funcionamento, bem como a descrição de cada uma das suas etapas. Em sequência, a estrutura geral dos EAs para otimização multiobjetivo é descrita, bem como técnicas para avaliar o desempenho dos MOEAs. Por fim, apresenta-se a técnica evolucionária de programação genética que é amplamente utilizada nesta tese.

4.1 Fundamentos

A EC consiste em uma gama de algoritmos estocásticos baseados na teoria da seleção natural de Darwin (SRINIVAS e DEB, 1994). Em um conjunto de indivíduos, aqueles que apresentam maior adaptabilidade ao ambiente considerado possuem maior expectativa de se manterem vivos e gerarem descendentes. Enquanto os menos aptos tendem a desaparecer e não perpetuarem suas características para as gerações futuras (DARWIN, 1859).

Os EAs funcionam de acordo com os processos evolutivos da natureza. Manipulam um conjunto de indivíduos (possíveis soluções para um problema de otimização) e operam no material genético dos indivíduos (genótipo, que é a representação codificada das potenciais soluções) por meio de operações genéticas de cruzamento e mutação que recombinam esse material. Para orientar o processo evolutivo, associa-se a cada indivíduo um valor de mérito (*fitness*) que reflete o quão apto ele é em relação aos outros. Quanto mais elevado o mérito de um indivíduo, maior será sua chance de sobreviver, reproduzir e transmitir suas características para as próximas gerações. A estrutura geral de funcionamento dos EAs é descrita a seguir (BACK *et al.*, 1997):

Algoritmo 4-1– Estrutura geral dos algoritmos evolucionários.

```
1.  g = 0.
2.  Inicie a população (g).
3.  Avalie a população (g).
4.  Enquanto o critério de parada não for satisfeito
5.  {
6.  nova população(g) = variação da população(g).
7.  Avalie a nova população(g).
8.  população (g + 1) = seleção {nova população(g)  U  Q}.
9.  g = g +1.
10. }
```

No Algoritmo 4-1, g indica a geração, **Q** é um conjunto de indivíduos que podem ser considerados na seleção, como por exemplo, a população(g), no entanto, **Q** também pode ser o conjunto vazio. A cada iteração uma nova população é gerada por meio de uma variação (recombinação dos indivíduos) da população atual. Os indivíduos da nova população são avaliados e o mecanismo de seleção guia o algoritmo para buscar melhores soluções. O processo de evolução continua até que algum critério de parada seja satisfeito. Detalhes específicos sobre a técnica evolucionária mais conhecida (algoritmo genético) como representação dos indivíduos, mecanismos de seleção, procedimentos de

recombinação, ajuste de parâmetros de controle, entre outros podem ser consultados em (VASCONCELOS *et al.*, 2001), (SOARES, 1997), (WHITLEY, 2001), (DIAS, 2000), (SRINIVAS e DEB, 1994), (BACK *et al.*, 1997) e (EIBEN *et al.*, 1999).

4.2 Algoritmos Evolucionários Multiobjetivo

Como visto anteriormente, para resolver o MOP um algoritmo necessita encontrar não uma única solução, mas sim um conjunto de soluções não-dominadas. Diante desse fato, percebe-se que os EAs são apropriados ao MOP. Afinal, enquanto os algoritmos de otimização determinísticos retornam uma única solução no término do processo de otimização, os EAs trabalham com uma população de soluções candidatas. Com isso, em uma única execução pode-se obter um grande número de soluções não-dominadas. Além disso, os EAs são capazes de lidar com dificuldades como a não convexidade e/ou descontinuidade da fronteira Pareto ótima (COELLO, 2006).

Segundo COELLO *et al.* (2007) as principais metas que os MOEAs devem atender são:

1. Manter as soluções não-dominadas ao longo das gerações.
2. Obter uma fronteira de soluções admissíveis não-dominadas o mais próxima possível da fronteira Pareto ótima.
3. Fornecer diversidade de pontos na melhor fronteira obtida e/ou no melhor conjunto de soluções eficientes encontrado.

A Tabela 4-1 exibe as principais técnicas implementadas pelos MOEAs a fim de atender a cada uma dessas metas. Para mais detalhes das técnicas consulte (COELLO *et al.*, 2007).

Tabela 4-1– Principais técnicas utilizadas pelos MOEAs para atender a cada meta.

Meta 1	R*anking* baseado em critérios de dominância e elitismo.
Meta 2	Geração de pontos não-dominados no espaço dos objetivos e incorporação de busca local.
Meta 3	*Fitness sharing*/técnica de nicho, distância da multidão e *clustering*.

O Algoritmo 4-2 representa a estrutura geral de um MOEA (COELLO *et al.*, 2007).

Algoritmo 4-2– Estrutura geral de um MOEA.

1. Crie a população P com n indivíduos;
2. Avalie a população P.
3. Classifique P com base em critérios de dominância e distância de multidão.
4. Enquanto o critério de parada não for satisfeito
5. **{**
   - 1.1. Execute processo de seleção em P e armazene os indivíduos selecionados em P^i.
   - 1.2. Gere nova população de pontos P^i.
   - 1.3. Avalie P^i.
   - 1.4. Junte as populações de pais e filhos (P U P^i) e classifique a população resultante segundo os critérios de dominância e distância de multidão.
   - 1.5. Armazene os n primeiros indivíduos em P.
6. **}**

Diversos MOEAs são propostos na literatura, por exemplo: *Nondominated Sorting Genetic Algorithm* (NSGA), *Niched-Pareto Genetic Algorithm* (NPGA), *Strength Pareto Evolutionary Algorithm* (SPEA), *Strength Pareto Evolutionary Algorithm 2* (SPEA2), *Pareto Archived Evolution Strategy* (PAES), *Nondominated Sorting Genetic Algorithm II* (NSGA-II) e *Micro-Genetic Algorithm for Multiobjective Optimization* (μGA). Consulte (COELLO, 2006) e (COELLO *et al.*, 2007) para obter informações específicas sobre esses algoritmos.

4.3 Comparação de Desempenho dos MOEAs

Os MOEAs não garantem que a melhor fronteira de soluções não-dominadas retornadas por eles seja a fronteira Pareto ótima (FPO). Portanto, tem-se como retorno a melhor aproximação possível da FPO, denominada melhor fronteira obtida (MFO). Dessa forma, faz-se necessário avaliar a qualidade da MFO a fim de verificar o quão bem um dado MOEA resolve determinado MOP. Por meio dessa avaliação podem-se identificar potencialidades e fraquezas de cada algoritmo. A questão é: como avaliar o desempenho dos MOEAs?

Deve ser observado que para mensurar a qualidade dos resultados devem-se levar em conta os recursos utilizados na obtenção da MFO. Alguns critérios comumente adotados são o tempo de execução e o número de avaliações do conjunto de funções que descrevem o modelo matemático do problema (ZITZLER *et al.*, 2003). Além desses indicadores simples, de acordo com COELLO *et al.* (2007) os aspectos gerais para medir o desempenho dos MOEAs incluem as *relações de dominância*, as *Empirical Attainment Functions* (EAFs) e os *indicadores de qualidade* sobre a MFO. Detalhes sobre as EAFs podem ser encontrados em (FONSECA e FLEMING, 1996) e (FONSECA *et al.*, 2005). Quanto aos indicadores de qualidade, têm-se: taxa de erro, cardinalidade, *coverage relationship*, *generational distance*, indicador de diversidade, *spread*, hipervolume, entre outros. Para explicação detalhada dos indicadores de qualidade consulte (VELDHUIZEN e LAMONT, 1999), (COELLO *et al.*, 2007), (ZITZLER e THIELE, 1999), (SRINIVAS e DEB, 1994), (SCHOTT, 1995), (KNOWLES, 2002), (ZITZLER e THIELE, 1998) e (HANSEN e JASZKIEWICZ, 1998). As relações de dominância são apresentadas na Tabela 4-2 que é adaptada de COELLO *et al.* (2007). Para facilitar a compreensão considere um MOP com n_f objetivos a serem minimizados no qual o vetor das funções objetivo é representado por $\boldsymbol{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_{n_f}(\mathbf{x}))^T$ pertencente ao espaço dos objetivos ($\mathbf{Z}$).

Tabela 4-2- Relações de dominância entre os vetores no espaço dos objetivos.

Relação	**Vetor pertencente ao espaço dos objetivos ($f(x)$)**	
Domina estritamente	$f_i(\mathbf{x}_1) \prec\prec f_i(\mathbf{x}_2)$	$\forall_{i\in[1,n_f]}$, $f_i(\mathbf{x}_1)$ é melhor que $f_i(\mathbf{x}_2)$
Domina	$f_i(\mathbf{x}_1) \prec f_i(\mathbf{x}_2)$	$\forall_{i\in[1,n_f]}$, $f_i(\mathbf{x}_1)$ não é pior que $f_i(\mathbf{x}_2)$ em todos os objetivos e é melhor em ao menos um.

Domina fracamente	$f_i(\mathbf{x}_1) \preccurlyeq f_i(\mathbf{x}_2)$	$\forall_{i\in[1,n_f]}$, $f_i(\mathbf{x}_1)$ *não é pior que* $f_i(\mathbf{x}_2)$ em todos os objetivos
Incomparável	$f_i(\mathbf{x}_1) \parallel f_i(\mathbf{x}_2)$	$\forall_{i\in[1,n_f]}$, $f_i(\mathbf{x}_1)$ não domina fracamente $f_i(\mathbf{x}_2)$ *e* $f_i(\mathbf{x}_2)$ não domina fracamente $f_i(\mathbf{x}_1)$
Indiferente	$f_i(\mathbf{x}_1) \sim f_i(\mathbf{x}_2)$	$\forall_{i\in[1,n_f]}$,$f_i(\mathbf{x}_1)$ tem o mesmo valor de $f_i(\mathbf{x}_2)$ para todos os objetivos

Como exemplo, a Figura 4-1 mostra graficamente as seguintes relações de dominância aplicada aos vetores em **Z**: $\mathbf{k} \prec \mathbf{l}, \mathbf{k} \prec \mathbf{m}, \mathbf{k} \prec \mathbf{n}, \mathbf{l} \prec \mathbf{n}, \mathbf{m} \prec \mathbf{n}, \mathbf{k} \prec\prec \mathbf{n}, \mathbf{k} \preccurlyeq \mathbf{k}, \mathbf{k} \preccurlyeq \mathbf{l}, \mathbf{k} \preccurlyeq \mathbf{m}, \mathbf{k} \preccurlyeq \mathbf{n}, \mathbf{l} \preccurlyeq \mathbf{l}, \mathbf{l} \preccurlyeq \mathbf{n}, \mathbf{m} \preccurlyeq \mathbf{m}, \mathbf{m} \preccurlyeq \mathbf{n}, \mathbf{n} \preccurlyeq \mathbf{n}$ e $\mathbf{l} \parallel \mathbf{m}$.

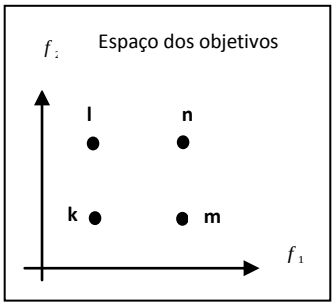


Figura 4-1 – Exemplo das Relações de dominância entre os vetores no espaço dos objetivos.

Dada as relações de dominância entre os vetores é possível formalizar o conceito de conjunto de soluções não-dominadas conforme ZITZLER *et al.* (2003), em termos do espaço dos objetivos. Este conjunto representa o retorno de um MOEA e consiste em um conjunto de soluções incomparáveis.

Definição 4-1 *Seja* $\mathbf{A} \subseteq \mathbf{Z}$ *um conjunto de vetores objetivo.* **A** *é dito um conjunto Pareto-Ótimo aproximado ou conjunto de soluções não-dominadas se qualquer vetor de* **A** *não domina fracamente qualquer outro de* **A**.

De acordo com COELLO *et al.* (2007) as relações de dominância também são aplicadas aos conjuntos de soluções não-dominadas conforme apresentado na Tabela 4-3.

Tabela 4-3 - Relações de dominância entre conjuntos de soluções não-dominadas no espaço dos objetivos.

Relação	**Conjunto de Soluções Não-dominadas**	
Domina estritamente	$\mathbf{A} \prec\prec \mathbf{B}$	Todo $\mathbf{x}_2 \in \mathbf{B}$ é estritamente dominado por ao menos um $\mathbf{x}_1 \in \mathbf{A}$
Domina	$\mathbf{A} \prec \mathbf{B}$	Todo $\mathbf{x}_2 \in \mathbf{B}$ é dominado por ao menos um $\mathbf{x}_1 \in \mathbf{A}$
Melhor	$\mathbf{A} \lhd \mathbf{B}$	Todo $\mathbf{x}_2 \in \mathbf{B}$ é fracamente dominado por ao menos um $\mathbf{x}_1 \in \mathbf{A}$ *e* $\mathbf{A} \neq \mathbf{B}$
Domina fracamente	$\mathbf{A} \preccurlyeq \mathbf{B}$	Todo $\mathbf{x}_2 \in \mathbf{B}$ é fracamente dominado por ao menos um $\mathbf{x}_1 \in \mathbf{A}$
Incomparável	$\mathbf{A} \parallel \mathbf{B}$	**A** não domina fracamente **B** e **B** não domina fracamente **A**
Indiferente	$\mathbf{A} \sim \mathbf{B}$	**A** domina fracamente **B** *e* **B** domina fracamente **A**

Na Figura 4-2 extraída de (ZITZLER *et al.*, 2003) são exemplificadas as relações de dominância entre três conjuntos de soluções não-dominadas, são elas: $\mathbf{A_1} \prec \mathbf{A_3}$, $\mathbf{A_2} \prec \mathbf{A_3}$, $\mathbf{A_1} \prec\prec \mathbf{A_3}$, $\mathbf{A_1} \preccurlyeq \mathbf{A_1}$, $\mathbf{A_1} \preccurlyeq \mathbf{A_2}$, $\mathbf{A_1} \preccurlyeq \mathbf{A_3}$, $\mathbf{A_2} \preccurlyeq \mathbf{A_2}$, $\mathbf{A_2} \preccurlyeq \mathbf{A_3}$, $\mathbf{A_3} \preccurlyeq \mathbf{A_3}$, $\mathbf{A_1} \lhd \mathbf{A_2}$, $\mathbf{A_1} \lhd \mathbf{A_3}$ e $\mathbf{A_2} \lhd \mathbf{A_3}$.

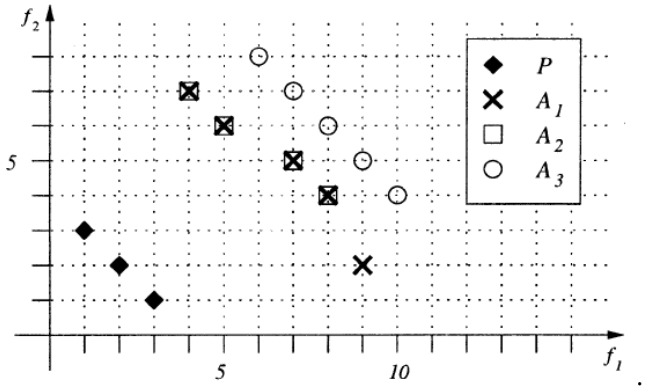


Figura 4-2 - Fronteira Pareto ótima **P** e os conjuntos de soluções não-dominadas $\mathbf{A}_1$, $\mathbf{A}_2$ e $\mathbf{A}_3$ de um MOP.

Mensurar o desempenho de MOEAs por meio da comparação de dominância entre os conjuntos de soluções não-dominadas tem a vantagem de não possuir nenhum viés em relação às preferências do decisor. No entanto, não fornece muitas informações sobre a solução obtida e, além disso, há situações em que não é possível determinar o melhor entre dois conjuntos de soluções não-dominadas (COELLO *et al.*, 2007).

4.4 Programação Genética

Nesta seção, inicialmente, apresentam-se os fundamentos da Programação Genética (*Genetic Programming*- GP). Em seguida, descreve-se com mais detalhes sua estrutura de funcionamento e cada uma de suas etapas.

4.4.1 Fundamentos

Na década de 50, Arthur Samuel levantou a seguinte questão: como computadores podem aprender a resolver problemas para os quais não foram explicitamente programados? De acordo com KOZA (1992), um entrave ao buscar soluções para problemas dessa natureza é a estrutura da solução. As estruturas baseadas em vetores de pesos, árvores de decisão, regras de produção, *strings* de cromossomos, as quais são utilizadas em metodologias como: aprendizado de máquina, inteligência artificial, redes neurais, entre outras, não possuem flexibilidade adequada para programação de computadores. KOZA (1992) defende a ideia de que a estrutura das soluções para um determinado problema é consequência da aptidão dos indivíduos ao longo do tempo. E afirma que, se o objetivo é usar o computador para solucionar problemas para os quais não foi explicitamente programado deve-se utilizar programas de computador como estrutura.

KOZA (1992) descreve a GP como uma técnica em computação evolucionária para aprendizagem automatizada de programas de computador que objetiva resolver problemas complexos nos mais distintos campos do conhecimento.

Conforme mostrado no Algoritmo 4-3 o processo de solucionar um problema via GP consiste em gerar aleatoriamente uma população inicial de programas de computador com tamanho e formato variados. Cada programa de computador recebe um valor de *fitness* de acordo com sua capacidade de solucionar o problema. Em seguida, conforme os valores dos parâmetros de entrada, aplicam-se operações genéticas aos programas de computador selecionados com base em seus valores de *fitness*. Depois de um período de gerações, têm-se programas de computador que melhor resolvem o problema em questão (POLI *et al.*, 2008).

Algoritmo 4-3– Algoritmo de programação genética.

1. Crie aleatoriamente uma população inicial de programas de acordo com o conjunto primitivo adequado ao domínio do problema.
2. Repita até que uma solução aceitável seja encontrada ou algum critério de parada seja satisfeito
   2.1. Execute cada programa e determine o valor da função *fitness.*
   2.2. Execute o processo de seleção sobre os programas da população para participar das operações genéticas.
   2.3. Gere novos programas a partir da execução das operações genéticas.
3. Retorne o melhor programa encontrado.

Em GP os indivíduos são representados por programas de computador, os quais podem mudar de forma e tamanho dinamicamente ao longo do processo de busca por soluções. O conjunto de todas as possíveis estruturas em GP equivale a todas as possíveis combinações que se pode obter com um conjunto primitivo (PS), o qual é composto pelo conjunto de funções (FS) união com o conjunto de terminais (TS). Matematicamente, tem-se:

$$\begin{aligned} \mathbf{FS} &= \{f_1, f_2, \dots, f_{Nfunc}\}, \\ \mathbf{TS} &= \{t_1, t_2, \dots, t_{Nterm}\}, \\ \mathbf{PS} &= \mathbf{FS} \cup \mathbf{TS}, \end{aligned} \tag{4-1}$$

sendo $Nfunc$ a quantidade de funções e $Nterm$ a quantidade de terminais.

No conjunto **FS** podem-se encontrar funções aritméticas, funções matemáticas, operações booleanas, operadores condicionais, operadores relacionais, comandos de decisão, comandos de repetição, entre outros adequados ao domínio do problema em questão. Convém ressaltar que cada elemento do conjunto **FS** possui um número de argumentos associado, o qual é chamado de aridade da função. O **TS** é tipicamente constituído por variáveis, constantes numéricas e funções com aridade zero (KOZA, 1992). Há diversos problemas para os quais não se pode ter um programa de computador como estrutura. Nesses casos, o conjunto primitivo deve ser composto por elementos relacionados ao domínio desse tipo de problema (POLI *et al.*, 2008).

Um programa em GP ao invés de ser formado por linhas de código é usualmente representado utilizando-se a estrutura de dados em árvore. Como se pode observar na Figura 4-3, extraída de (POLI

et al., 2008), os nós internos da árvore são formados por elementos pertencentes ao **FS**, os nós folhas são constituídos por elementos do **TS**. Esta figura representa o programa $\max(x + x, x + 3 * y)$.

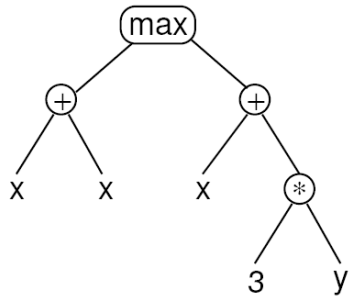


Figura 4-3 – Representação de um indivíduo utilizando a estrutura de árvore.

Em GP os conjuntos FS e TS devem ser estabelecidos de modo a satisfazer as propriedades de fechamento e suficiência (POLI *et al.*, 2008). Para satisfazer a propriedade de fechamento é necessário que haja consistência de tipos entre a entrada e saída de parâmetros nas funções. Isso significa que cada uma das funções do FS deve ser capaz de receber em seu(s) argumento(s) qualquer valor possível de ser retornado por qualquer outra função ou assumido por qualquer terminal. Outra exigência da propriedade de fechamento é a avaliação de segurança, que está relacionada às funções que podem falhar em tempo de execução. Por exemplo, uma divisão por zero pode ser produzida em algum programa. Nesse caso, uma forma de tratamento é a construção de funções protegidas, as quais tratam essas exceções (suponha a existência da função pdiv que retorna 1 como resultado se o denominador for zero). Outra possibilidade é penalizar o *fitness* dos programas inválidos. No entanto, se existirem muitos programas nessa situação a seleção ficará prejudicada (KOZA, 1992). No que se refere à propriedade de suficiência, caso seja possível expressar uma solução válida para o problema tratado utilizando somente elementos existentes no conjunto primitivo **PS**, pode-se concluir que a suficiência está assegurada.

Em relação à construção da população inicial, essa é gerada de forma aleatória. No entanto, existem algumas estratégias para nortear esse procedimento. KOZA (1992) apresenta três técnicas básicas: *full*; *grow* e *ramped half-and-half*. As três técnicas exigem que se estabeleça uma altura máxima para as árvores geradas. A altura de um nó é definida pela quantidade de nós que devem ser percorridos do nó corrente até o nó raiz (para o qual é assumida a altura zero). A altura de uma árvore é a altura do nó de maior altura existente na árvore.

Na técnica *full* as árvores construídas devem possuir altura igual à altura máxima estabelecida. Para que se possa garantir essa condição, basta restringir a escolha dos nós da árvore da seguinte forma: se após a inserção do nó a altura da árvore é menor que a altura máxima, o nó selecionado deve pertencer ao **FS**. Caso contrário, ou seja, a altura da árvore é igual à altura máxima, deve-se escolher um nó pertencente ao **TS**.

Na técnica *grow* a altura da árvore gerada deve ser igual ou menor a altura máxima estabelecida. Para garantir essa condição basta adicionar nós pertencentes ao **PS** na árvore, desde que a altura da árvore não supere a altura máxima. Caso a inserção de um nó leve a árvore a atingir a altura máxima, o nó a ser inserido deve pertencer ao **TS**.

A técnica *ramped half-and-half* consiste na combinação dos métodos *full* e *grow* do seguinte modo: i) estabelecem-se diversas faixas de altura (por exemplo, a partir de 2) até a altura máxima; ii) dividi-se o total de indivíduos da população pela quantidade de faixas existentes a fim de saber qual a quantidade de árvores que devem ser geradas para cada altura; e iii) cria-se, para cada uma das faixas de altura, metade da população usando *full* e a outra metade usando *grow*. A técnica *ramped half-and-half*, em geral é mais usada, pois assegura que as árvores geradas terão formatos e alturas variadas. Dessa forma, atende a uma considerável gama de problemas.

4.4.2 Função de *Fitness*

Na GP faz-se necessário atribuir ao indivíduo um valor de acordo com sua capacidade de resolver o problema em questão. Esse procedimento é conhecido como função de avaliação ou *fitness*. Em GP, para determinar a *fitness* é comum utilizar os chamados *fitness cases*, os quais representam um conjunto finito de situações distintas existentes no domínio de busca.

Conforme KOZA (1992), há quatro medidas de *fitness* que são comumente utilizadas, as quais são descritas, a seguir:

i). *Raw fitness*: refere-se a uma medida de desempenho, a qual é estabelecida de acordo com a natureza do problema tratado. Matematicamente é expressa por:

$$r_{i,t} = \sum_{j=1}^{N} \left| S_{i,j} - C_j \right|, \tag{4-2}$$

sendo que t indica o tempo geracional, $S_{i,j}$ o valor obtido para o i-ésimo programa considerando-se o j-ésimo *fitness case* e C_j o valor correto para o j-ésimo *fitness case*. Em geral, inclusive neste trabalho, o *raw fitness* está relacionado ao conceito de erro. Neste caso, subentende-se que quanto menor o seu valor, melhor será o programa.

ii).*Standardized fitness*: consiste em estabelecer uma medida de desempenho na qual quanto menor o valor, maior a qualidade do indivíduo. Para os casos nos quais essa situação já ocorre no *raw fitness*, por exemplo, quando este se refere ao erro, tem-se que:

$$s_{i,t} = r_{i,t}, \tag{4-3}$$

sendo $s_{i,t}$ o *standardized fitness* do i-ésimo programa no tempo geracional t.

Para os outros casos, quando é desejável que o melhor indivíduo possua valor de aptidão zero, faz-se necessário subtrair ou adicionar uma constante. Por exemplo, para um problema em que o melhor indivíduo possui um valor maior de aptidão, basta calcular o *standardized fitness* do seguinte modo:

$$s_{i,t} = r_{max} - r_{i,t}, \tag{4-4}$$

sendo $s_{i,t}$ o *standardized fitness* do i-ésimo programa no tempo geracional t, $r_{i,t}$ o *raw fitness* do i-ésimo programa no tempo geracional t e r_{max} o maior valor possível para o *raw fitness*.

iii). *Adjusted fitness*: consiste em aplicar um ajuste ao *standardized fitness* de modo que o valor obtido fique compreendido entre zero e um. Além disso, quanto melhor o indivíduo maior o seu valor de aptidão. Matematicamente o *adjusted fitness* é dado por:

$$a_{i,t} = \frac{1}{1 + s_{i,t}}, \tag{4-5}$$

sendo $s_{i,t}$ o *standardized fitness* do i-ésimo programa no tempo geracional t.

iv). *Normalized fitness*: consiste em tornar o valor de aptidão proporcional para cada programa considerando-se toda a população. É calculado a partir do *adjusted fitness* da seguinte forma:

$$n_{i,t} = \frac{a_{i,t}}{\sum_{k=1}^{M} a_{k,t}}, \tag{4-6}$$

sendo $a_{i,t}$ o *adjusted fitness* do i-ésimo programa no tempo geracional t e M o total de programas na população.

4.4.3 Operações Genéticas

As principais operações genéticas realizadas em GP são seleção, cruzamento, mutação e reprodução. A reprodução é direta e consiste simplesmente em selecionar um percentual previamente definido de indivíduos da população atual baseados no valor de *fitness* desses e copiá-los para a próxima geração. Em relação à seleção, cruzamento e mutação existem várias formas de realizar cada uma dessas operações genéticas, mais detalhes podem ser encontrados em POLI *et al.* (2008). A seguir, apresentam-se as formas utilizadas nesta tese: seleção por torneio, o cruzamento denominado *subtree crossover* e a mutação denominada *node replacement mutation.*

Na seleção por torneio, alguns programas (pelo menos dois) são escolhidos aleatoriamente na população. Todos os programas escolhidos são comparados entre si e aquele com melhor valor de *fitness* é selecionado.

No cruzamento selecionam-se dois programas. Escolhe-se aleatoriamente um nó em cada um dos programas, tendo como única restrição o seguinte: os nós selecionados devem pertencer ao mesmo conjunto (seja de funções ou terminais). Com isso, obtêm-se duas sub-árvores, uma para cada programa. A partir dos programas pais são gerados dois programas filhos. Isso é feito trocando-se as sub-árvores de cada um dos pais. Esse procedimento denominado *subtree crossover* é mostrado na Figura 4-4 (os nós em negrito são escolhidos aleatoriamente e representam o ponto de corte de cada programa).

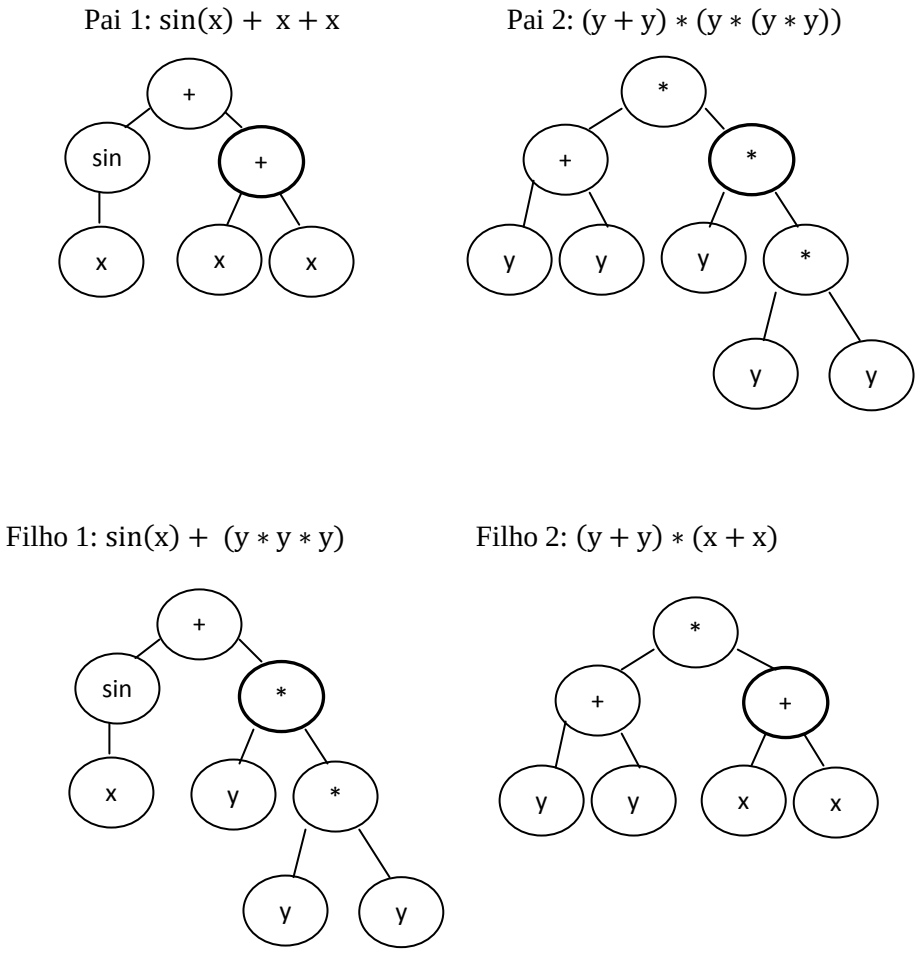


Figura 4-4 – Exemplo de cruzamento.

Na mutação, a partir da escolha de um programa, seleciona-se aleatoriamente um nó que é substituído por outro nó escolhido por sorteio. A restrição imposta é que o nó sorteado deve pertencer ao mesmo conjunto (seja de funções ou terminais) do nó selecionado. Além disso, no caso de funções, tanto o nó selecionado quanto o sorteado devem possuir a mesma aridade. Essa mutação é chamada de *node replacement mutation* e é exemplificada na Figura 4-5 (o nó selecionado está em negrito).

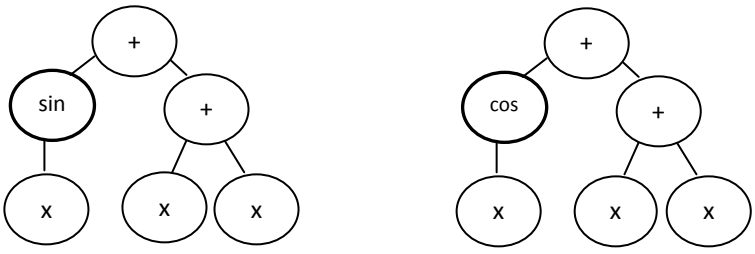


Figura 4-5 – Exemplo de mutação.

4.5 Conclusão

Neste capítulo, primeiramente, foram apresentados os conceitos fundamentais envolvendo os EAs. Em seguida, fez-se a descrição da estrutura geral dos MOEAs e das principais formas de mensurar o

desempenho desses para fins de comparação. Por fim, a GP e seus principais detalhes foram descritos. Existem diversas questões na área de EC a serem exploradas, por exemplo, formas adequadas de comparação entre as melhores fronteiras obtidas por cada algoritmo, como construir MOEAs para lidar com problemas com incertezas, entre outros.

5. ANÁLISE INTERVALAR

A análise intervalar (IA) emprega uma aritmética própria definida sobre conjuntos de intervalos reais, diferentemente dos métodos tradicionais que, em geral, utilizam o conjunto de número reais. Formas de se trabalhar com IA apareceram em pesquisas dos anos de 1924 e 1931 (BURKILL, 1924; YOUNG, 1931). No trabalho de SUNAGA (1958), importantes contribuições foram propostas. Foi a partir da tese de Moore em 1962, e principalmente do seu livro que o assunto despertou maior atenção dos pesquisadores (MOORE, 1962; MOORE, 1966).

De acordo com RUESTSCH (2005), os métodos intervalares cresceram devido à motivação de se controlar erros de arredondamento nas operações computacionais ponto-flutuante. Como os tipos de dados das linguagens de programação representam apenas uma abstração (subconjunto finito) para determinado conjunto numérico, pode existir erro se um valor não puder ser representado exatamente. A matemática intervalar[8] propõe representar o valor por um limite inferior e um limite superior, ou seja, um intervalo. MOORE *et al.* (2009) afirmam que por meio da computação intervalar pode-se construir programas para encontrar intervalos que contêm com absoluta certeza a resposta exata para vários problemas matemáticos. Os limites extremos do intervalo podem não ser representáveis com precisão em máquina. Nesse caso, utiliza-se um procedimento denominado *outward rounding*, o qual consiste em arredondar o limite inferior para o primeiro número representável em máquina inferior a ele, analogamente o limite superior é arredondado para o primeiro número superior a ele representável em máquina. Este procedimento é importante porque fica garantido que o intervalo resultante de operações aritméticas intervalares esteja contido entre os limites extremos obtidos pelo *outward rounding*.

Conforme JAULIN *et al.* (2001), os algoritmos intervalares são mais complexos do que os algoritmos não intervalares destinados a resolver um mesmo problema. Algumas vezes, os algoritmos intervalares demandam por subdivisão de intervalos, o que resulta em sobrecarga computacional. Sendo assim, em geral, os algoritmos intervalares podem gastar mais tempo de processamento e memória. Como vantagem, os algoritmos baseados em intervalos podem tratar classes mais gerais de problemas e garantir a confiabilidade do resultado.

Atualmente, a IA é aplicada em diversas áreas, entre as quais: robótica, controle robusto e problemas de estimativa (JAULIN *et al.*, 2001). No caso de controle robusto, o termo robustez pode ser associado à incerteza existente na modelagem do processo a ser controlado, o que implica diretamente na confiabilidade do sistema diante de uma eventual falha. Nesse tipo de problema a IA é aplicada dada a alta capacidade que possui para tratar as incertezas presentes.

[8] Os termos matemática intervalar e análise intervalar são utilizados como sinônimos nesse trabalho.

5.1 Noções da Teoria de Conjuntos

A computação intervalar é um caso particular da computação sobre conjuntos e a teoria de conjuntos fornece os fundamentos para a IA. A seguir, apresenta-se uma revisão de teoria de conjuntos fundamentada em JAULIN *et al.* (2001).

Considere dois conjuntos **X** e **Y**. A interseção é definida por

$$\mathbf{X} \cap \mathbf{Y} \triangleq \{x | x \in \mathbf{X} \wedge x \in \mathbf{Y}\}, \tag{5-1}$$

e a união é

$$\mathbf{X} \cup \mathbf{Y} \triangleq \{x | x \in \mathbf{X} \vee x \in \mathbf{Y}\}, \tag{5-2}$$

e o complemento relativo de **Y** em **X** é

$$\mathbf{X} \backslash \mathbf{Y} \triangleq \{x | x \in \mathbf{X} \wedge x \notin \mathbf{Y}\}, \tag{5-3}$$

e o produto cartesiano de **X** e **Y** é

$$\mathbf{X} \times \mathbf{Y} \triangleq \{(x, y) | x \in \mathbf{X} \wedge y \in \mathbf{Y}\}. \tag{5-4}$$

Se $\mathbf{Z} = \mathbf{X} \times \mathbf{Y}$, a projeção de um subconjunto $\mathbf{Z}_1$ de **Z** sobre **X** (com respeito a **Y**) é definida como

$$proj_{\mathbf{X}} \triangleq \{x \in \mathbf{X} | \exists\, y \in \mathbf{Y} | (x, y) \in \mathbf{Z}_1\}. \tag{5-5}$$

A Figura 5-1 extraída de SOARES (2008) ilustra as definições (5-4) e (5-5).

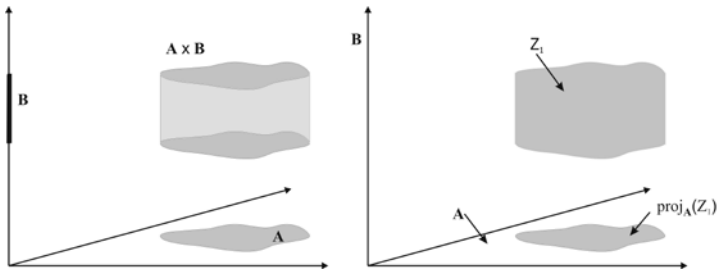


Figura 5-1 - Operações sobre conjuntos.

A inclusão de **X** em **Y** é definida por

$$\mathbf{X} \subset \mathbf{Y} \Leftrightarrow \forall \mathbf{x} \in \mathbf{X}, \mathbf{x} \in \mathbf{Y}, \tag{5-6}$$

e a igualdade de **X** e **Y** por

$$\mathbf{X} = \mathbf{Y} \Leftrightarrow \mathbf{X} \subset \mathbf{Y} \wedge \mathbf{Y} \subset \mathbf{X}. \tag{5-7}$$

Considere uma função $f: \mathbf{X} \rightarrow \mathbf{Y}$. Se $\mathbf{X_1} \subset \mathbf{X}$ a, imagem direta de $\mathbf{X_1}$ por f é

$$f(\mathbf{X_1}) \triangleq \{f(x) | x \in \mathbf{X_1}\}. \tag{5-8}$$

Se $\mathbf{Y_1} \subset \mathbf{Y}$, a imagem recíproca de $\mathbf{Y_1}$ por f é

$$f^{-1}(\mathbf{Y_1}) \triangleq \{x \in \mathbf{X} | f(x) \in \mathbf{Y_1}\}. \tag{5-9}$$

Se $\emptyset$ denota o conjunto vazio então as definições anteriores implicam que

$$f(\emptyset) = f^{-1}(\emptyset) = \emptyset. \tag{5-10}$$

Segundo JAULIN *et al*. (2001), assumindo que $\mathbf{X}_1$ e $\mathbf{X}_2$ são subconjuntos de **X** e $\mathbf{Y}_1$ e $\mathbf{Y}_2$ são subconjuntos de **Y**, pode-se mostrar que:

$$
\begin{aligned}
&f(\mathbf{X_1} \cap \mathbf{X_2}) \subset f(\mathbf{X_1}) \cap f(\mathbf{X_2}),\\
&f(\mathbf{X_1} \cup \mathbf{X_2}) = f(\mathbf{X_1}) \cup f(\mathbf{X_2}),\\
&f^{-1}(\mathbf{Y_1} \cap \mathbf{Y_2}) = f^{-1}(\mathbf{Y_1}) \cap f^{-1}(\mathbf{Y_2}),\\
&f^{-1}(\mathbf{Y_1} \cup \mathbf{Y_2}) = f^{-1}(\mathbf{Y_1}) \cup f^{-1}(\mathbf{Y_2}),\\
&f(f^{-1}(\mathbf{Y})) \subset \mathbf{Y},\\
&f^{-1}(f(\mathbf{X})) \supset \mathbf{X},\\
&\mathbf{X_1} \subset \mathbf{X_2} \Longrightarrow f(\mathbf{X_1}) \subset f(\mathbf{X_2}),\\
&\mathbf{Y_1} \subset \mathbf{Y_2} \Longrightarrow f^{-1}(\mathbf{Y_1}) \subset f^{-1}(\mathbf{Y_2}).
\end{aligned}
\tag{5-11}
$$

5.2 Intervalos – Conceitos Básicos

Comumente na IA os números são representados por um limite inferior e um superior, formando assim um intervalo (RUETSCH, 2005). A seguir, são apresentadas algumas definições baseadas em (HICKEY *et al.*, 2001) e (JAULIN *et al.*, 2001).

Definição 5-1 *Um conjunto* $\mathbf{X}$ *de números reais é conexo se não existem conjuntos disjuntos não vazios* $\mathbf{X_1}$ *e* $\mathbf{X_2}$ *que o intersectam e para os quais* $\mathbf{X} \subset \mathbf{X_1} \cup \mathbf{X_2}$. *O conjunto vazio é conexo por definição.*

Definição 5-2 *Um intervalo fechado* $[\mathbf{x}]$ *é um subconjunto conexo de* $\mathbb{R}$ *definido por*

$$[\mathrm{x}] = [\mathrm{x}^-, \mathrm{x}^+] = \{\mathrm{x} \in \mathbb{R} \mid \mathrm{x}^- \leq \mathrm{x} \leq \mathrm{x}^+\}, \tag{5-12}$$

sendo x^- *e* x^+*os limites inferior e superior do intervalo, respectivamente.*

Ressalta-se que neste trabalho o termo intervalo refere-se ao intervalo fechado, que é aquele no qual os limites extremos estão inclusos. O conjunto de todos os intervalos do tipo definido em (5-12) representa o conjunto dos números intervalares reais ($\mathbb{IR}$).

Definição 5-3 *A largura de* $[\mathrm{x}]$ *é*

$$\mathrm{w}([\mathrm{x}]) = \mathrm{x}^+ - \mathrm{x}^-. \tag{5-13}$$

Definição 5-4 *O centro de* $[\mathrm{x}]$ *é*

$$\mathrm{mid}([\mathrm{x}]) = \frac{\mathrm{x}^+ + \mathrm{x}^-}{2}. \tag{5-14}$$

Para exemplificar, seja $[\mathrm{x}] = [0,2]$ e $[\mathrm{y}] = [-1,1]$, logo: $\mathrm{w}([\mathrm{x}]) = 2$, $\mathrm{w}([\mathrm{y}]) = 2$, $\mathrm{mid}([\mathrm{x}]) = 1$ e $\mathrm{mid}([\mathrm{y}]) = 0$.

Com base nas definições 5-3 e 5-4 outra forma de representar os números na IA é dada por:

$$[\mathrm{x}] = \mathrm{mid}([\mathrm{x}]) + \left[-\frac{1}{2}\mathrm{w}(\mathrm{x}), \frac{1}{2}\mathrm{w}(\mathrm{x})\right] = \mathrm{mid}([\mathrm{x}]) + \frac{1}{2}\mathrm{w}(\mathrm{x})[-1,1]. \tag{5-15}$$

Nessa representação a ideia é descrever um número em termos de um valor central (mid) e uma incerteza na medida de $\pm\frac{\mathrm{w}}{2}$. Por exemplo, $[\mathrm{x}] = [0,2]$ pode ser escrito como $[\mathrm{x}] = 1 + 1[-1,1]$.

Como citado na seção anterior o conceito de operações entre conjuntos é importante para trabalhar com a matemática intervalar. A seguir, segundo MOORE *et al.* (2009) são definidas as operações de interseção e união para intervalos, além disso, descreve-se o conceito de intervalo degenerado.

Definição 5-5 *A interseção entre* [x] e [y] *é dada por*

$$[x] \cap [y] = \emptyset, \text{se } x^+ < y^- \vee y^+ < x^-$$
$$[x] \cap [y] = [\max(x^-, y^-), \min(x^+, y^+)], \text{caso contrário.} \quad (5\text{-}16)$$

Para exemplificar, seja $[x] = [0,2]$ e $[y] = [-1,1]$, então: $[x] \cap [y] = [0,1]$.

Quanto à união, é fácil perceber que se os dois intervalos não possuírem pontos em comum o resultado da operação resultará em conjuntos disjuntos, os quais não podem ser expressos como intervalo. Para solucionar essa questão, define-se o intervalo *hull*, obtido pelo operador $\underline{\cup}$, como mostrado a seguir.

Definição 5-6 *Dado* [x] *e* [y] *o intervalo hull é*

$$[x] \underline{\cup} [y] = [\min(x^-, y^-), \max(x^+, y^+)]. \quad (5\text{-}17)$$

Dessa forma, a união sempre resultará em um intervalo e satisfará o seguinte:

$$[x] \cup [y] \subseteq [x] \underline{\cup} [y]. \quad (5\text{-}18)$$

Como exemplo, seja $[x] = [-1,0]$ e $[y] = [1,2]$, tem-se que $[x] \cup [y]$ não é conexo, portanto não pode ser expresso como um intervalo, no entanto, $[x] \underline{\cup} [y] = [-1,2]$.

Um intervalo é dito degenerado se $x^- = x^+$, sendo assim, entende-se que esse tipo de intervalo contém um único número real (o escalar x).

5.2.1 Vetor Intervalar

Até o momento mostrou-se a representação intervalar para números escalares. No entanto, na prática tem-se que os problemas reais possuem natureza multidimensional. Diante disso, nesta seção, conforme JAULIN *et al.* (2001) e MOORE *et al*. (2009) são apresentadas algumas definições acerca de vetores intervalares.

Definição 5-7 *Um vetor intervalar, também chamado de caixa, é o produto cartesiano de n intervalos fechados, sendo definido por*

$$[\mathbf{x}] = [x_1] \times [x_2] \times \ldots \times [x_n]. \quad (5\text{-}19)$$

O conjunto de todas as caixas na dimensão n é denotado por $\mathbb{IR}^n$. Na Figura 5-2 exibe-se uma caixa $[\mathbf{x}]$ em $\mathbb{IR}^2$.

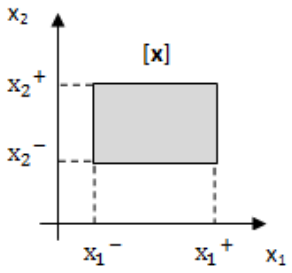


Figura 5-2- caixa $[\mathbf{x}]$ em $\mathbb{IR}^2$.

Definição 5-8 *O limite inferior* $\mathbf{x}^-$ *de* $[\mathbf{x}]$ *é dado por* $\mathbf{x}^- = (x_1^-, x_2^-, \ldots, x_n^-)^T$. Analogamente, *o limite superior* $\mathbf{x}^+ = (x_1^+, x_2^+, \ldots, x_n^+)^T$.

Definição 5-9 *A largura de* $[\mathbf{x}]$ é

$$w([\mathbf{x}]) = \max_{1\le i\le n} w([x_i]). \tag{5-20}$$

Definição 5-10 *O centro de* $[\mathbf{x}]$ é

$$\mathrm{mid}([\mathbf{x}]) = (\mathrm{mid}([x_1]), \mathrm{mid}([x_2]), \dots, \mathrm{mid}([x_n]))^T. \tag{5-21}$$

Definição 5-11 *O plano principal de* $[\mathbf{x}]$ *é o plano simétrico perpendicular ao seu maior lado. Assumindo-se que j é o índice que identifica o plano principal, j é definido como:*

$$j = \min\{ i \mid w([x_i]) = w([\mathbf{x}])\}. \tag{5-22}$$

Definição 5-12 *A bissecção regular consiste em dividir* $[\mathbf{x}]$ *em duas caixas simétricas separadas pelo seu plano principal. Matematicamente, obtêm-se:*

$$\begin{aligned} [\mathbf{x}]_1 &= [x_1] \times \dots \times \left[x_j^-, \frac{x_j^- + x_j^+}{2}\right] \times \dots \times [x_n], \\ [\mathbf{x}]_2 &= [x_1] \times \dots \times \left[\frac{x_j^- + x_j^+}{2}, x_j^+\right] \times \dots \times [x_n]. \end{aligned} \tag{5-23}$$

Como exemplo, tem-se que a bissecção de $[\mathbf{x}] = [1,2] \times [-1,3]$ resulta em $[\mathbf{x}]_1 = [1,2] \times [-1,1]$ e $[\mathbf{x}]_2 = [1,2] \times [1,3]$.

5.3 Computação Intervalar

Nesta seção são definidas as operações aritméticas e funções intervalares fundamentais.

5.3.1 Operações Aritméticas

Definição 5-13 *Dado que* $\odot$ *representa qualquer uma das quatro operações clássicas, então*

$$[x] \odot [y] = \{x \odot y \mid x \in [x] \text{ e } y \in [y]\}. \tag{5-24}$$

Em termos dos limites extremos, cada operação clássica para intervalos não vazios fechados é definida como:

$$[x] + [y] = [x^- + y^-, x^+ + y^+]. \tag{5-25}$$

$$[x] - [y] = [x^- - y^+, x^+ - y^-]. \tag{5-26}$$

$$\begin{aligned} &[x] * [y] = [\min(s), \max(s)], \text{sendo} \\ &s = \{x^- * y^-, x^- * y^+, x^+ * y^-, x^+ * y^+\}. \end{aligned} \tag{5-27}$$

$$[x]/[y] = [x] * (1/[y]),$$

Sendo

$$\begin{aligned} &1/[y] = \emptyset, && \text{se } [y] = [0,0]. \\ &1/[y] = [1/y^+, 1/y^-], && \text{se } 0 \notin [y]. \\ &1/[y] = [1/y^+, \infty], && \text{se } y^- = 0 \wedge y^+ > 0. \\ &1/[y] = [-\infty, 1/y^-], && \text{se } y^- < 0 \wedge y^+ = 0. \\ &1/[y] = [-\infty, \infty], && \text{se } y^- < 0 \wedge y^+ > 0. \end{aligned} \tag{5-28}$$

A seguir, considerando-se $[x] = [0,2]$ e $[y] = [1,2]$ as operações definidas em (5-25), (5-26), (5-27) e (5-28) são exemplificadas:

$$[x] + [y] = [1, 4].$$
$$[x] - [y] = [-2, 1].$$
$$[x] * [y] = [0, 4].$$
$$[x]/[y] = [0, 2].$$

Convém ressaltar que a propriedade distributiva não é válida na computação intervalar. No entanto, é satisfeita a subdistributiva que é dada por

$$[x] * ([y] + [z]) \subseteq [x] * [y] + [x] * [z]. \quad (5\text{-}29)$$

5.3.2 Funções Intervalares

Para determinar a imagem de uma função real $f(x): \mathbb{R} \to \mathbb{R}$, quando x varia em um respectivo intervalo [x], pode-se computar f para todos os possíveis valores do intervalo. Matematicamente, tem-se a função intervalar:

$$F([x]) = \{ f(x) | x \in [x] \}. \quad (5\text{-}30)$$

Segundo MOORE *et al*. (2009), a função intervalar (5-30) é fácil de ser calculada para algumas funções. Por exemplo, seja $f(x) = x^2, x \in \mathbb{R}$**.** Dado [x], tem-se que $F([x]) = \{x^2 | x \in [x]\}$, a qual pode ser expressa por

$$\begin{array}{l} F([x]) = [(x^-)^2, (x^+)^2], \text{ se } 0 \leq x^- \leq x^+. \\ F([x]) = [(x^+)^2, (x^-)^2], \text{ se } x^- \leq x^+ \leq 0. \\ F([x]) = [0, \max\{(x^-)^2, (x^+)^2\}], \text{ se } x^- < 0 < x^+. \end{array} \quad (5\text{-}31)$$

Assumindo $[x] = [-1,1]$ tem-se $F([x]) = [0,1]$**.** Note que o valor de $F([x])$, caso a expressão de cálculo utilizada fosse $[x] * [x]$, resultaria em $[-1,1]$**.** Perceba que expressões semelhantes na aritmética real levam a resultados distintos na computação intervalar. Nesse caso, o que ocorre é um intervalo superestimado, uma vez que $[0,1] \subseteq [-1,1]$**.** Isso acontece, pois, no produto $[x] * [x]$, a IA assume que o primeiro fator [x] varia independentemente do segundo fator [x]. Essa situação é conhecida como *problema da dependência*, o qual é detalhado na Subseção 5.3.4.

Nos casos em que a função real é monótona não decrescente (f cresce com o crescimento de x) tem-se que [x] é mapeado diretamente para (MOORE *et al.*, 2009):

$$F([x]) = [f(x^-), f(x^+)]. \quad (5\text{-}32)$$

Como exemplo tem-se as funções elementares: exponencial, logarítmica, raiz quadrada, entre outras, as duas últimas para $x > 0$.

Em (5-30) e (5-32) calculou-se a imagem de uma função real que recebe argumentos intervalares por meio da computação da imagem de cada $x \in [x]$. Outra forma de encontrar a imagem de uma função real f é aplicar a *extensão intervalar*, que consiste em aplicar diretamente a fórmula de f nos argumentos intervalares.

Conforme MOORE *et al.* (2009), uma função qualquer é definida por dois itens: I) um domínio sobre o qual ela atua; II) por uma regra que especifica como os elementos do domínio são mapeados.

O item II isoladamente é uma fórmula. A aplicação de argumentos intervalares à fórmula de uma função resulta em uma *extensão* da função. Interpreta-se que o domínio foi estendido para incluir intervalos não degenerados.

Definição 5-14 *Sendo f uma função real, uma função intervalar F é uma extensão intervalar de f se*

$$F([x_1, x_1], [x_2, x_2], \dots, [x_n, x_n]) = f(x_1, x_2, \dots, x_n). \quad (5\text{-}33)$$

A expressão (5-33) indica que se os argumentos de F são intervalos degenerados, o valor da imagem de F deve ser um intervalo degenerado idêntico ao valor da imagem de f.

5.3.3 Funções de Inclusão

De acordo com MOORE *et al.* (2009) são apresentadas algumas definições:

Definição 5-15 *A função intervalar $F([x_1], [x_2], \dots, [x_n])$ é uma inclusão isotônica se*

$$[y_i] \subseteq [x_i] \; para \; i = 1, \dots, n \Longrightarrow F([y_1], \dots, [y_n]) \subseteq F([x_1], \dots, [x_n]). \quad (5\text{-}34)$$

Definição 5-16 *Uma função intervalar racional é uma função intervalar cujos valores são definidos por uma sequência finita de operações aritméticas intervalares. Todas as funções intervalares racionais são inclusões isotônicas.*

O autor (MOORE, 1966 apud MOORE *et al.*, 2009, p.47) define o teorema fundamental da análise intervalar como:

Teorema 1 Se a função intervalar F é uma inclusão isotônica e uma extensão intervalar de uma função real f, então

$$f([x_1], \dots, [x_n]) \subseteq F([x_1], \dots, [x_n]). \quad (5\text{-}35)$$

Analogamente a (5-35), considerando $\boldsymbol{f}: \mathbb{R}^{n_v} \to \mathbb{R}^{n_f}$, JAULIN *et al.* (2001) definem $[\mathbf{f}]: \mathbb{IR}^{n_v} \to \mathbb{IR}^{n_f}$ como função de inclusão para $\boldsymbol{f}$ se

$$\forall [\mathbf{x}] \in \mathbb{IR}^{n_v}, \boldsymbol{f}([\mathbf{x}]) \subseteq [\mathbf{f}]([\mathbf{x}]). \quad (5\text{-}36)$$

Para ilustrar o conceito, assuma $\boldsymbol{f}: \mathbb{R}^2 \to \mathbb{R}^2$. Na Figura 5-3, verifica-se que a função de inclusão $[\mathbf{f}]$ é uma caixa envolvendo $\boldsymbol{f}([\mathbf{x}])$. Dessa forma, o conceito de função de inclusão auxilia na resolução de problemas nos quais $\boldsymbol{f}$ é descontínua (uma união de conjuntos disjuntos) ou não convexa (existem pontos de $\boldsymbol{f}([\mathbf{x}])$ tais que o segmento que os conecta não está dentro de $\boldsymbol{f}([\mathbf{x}])$). Em outras palavras, qualquer que seja a forma de $\boldsymbol{f}([\mathbf{x}])$, uma função de inclusão $[\mathbf{f}]([\mathbf{x}])$ de $\boldsymbol{f}$ resulta em uma caixa que a contém. Um dos propósitos da IA é fornecer, para uma grande classe de funções $\boldsymbol{f}$, funções de inclusão que possam ser resolvidas rapidamente e que $w([\mathbf{f}]([\mathbf{x}]))$ não seja muito largo.

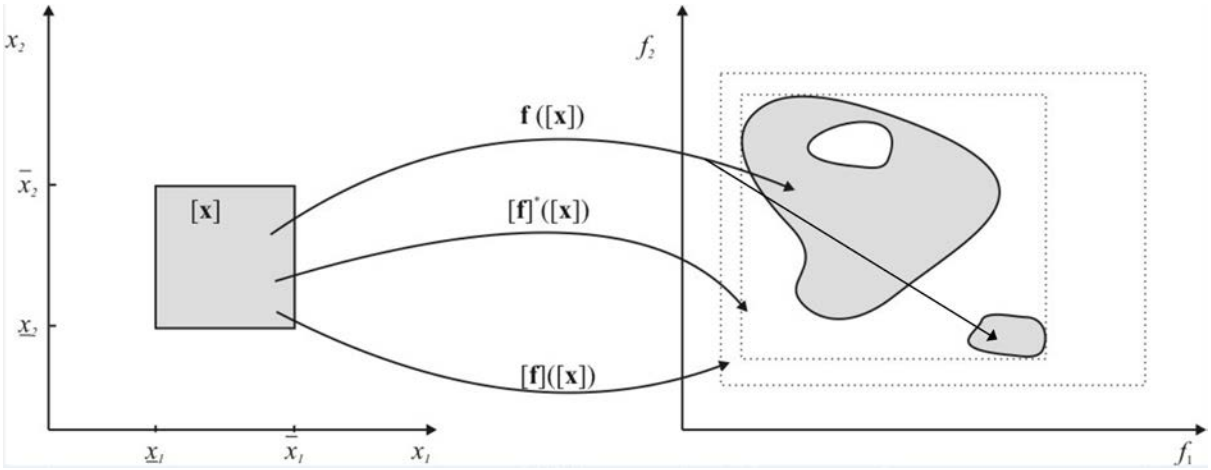


Figura 5-3- Imagem de uma caixa pela função **f** e duas de suas funções de inclusão [**f**] e [**f**]*. Sendo [**f**]* mínima. Adaptada de (SOARES, 2008).

Segundo JAULIN *et al*. (2001) uma função de inclusão pode ser estreita, convergente, monotônica e mínima. A seguir, esses conceitos são apresentados:

Estreita, se

$$\forall\, \mathbf{x} \in [\mathbf{x}], \qquad [\mathbf{f}](\mathbf{x}) = \boldsymbol{f}(\mathbf{x}). \tag{5-37}$$

Convergente, se

$$\lim_{k\to\infty} w([\mathbf{x}]_k) = 0 \implies \lim_{k\to\infty} w([\mathbf{f}]([\mathbf{x}]_k)) = 0. \tag{5-38}$$

Sendo que k indica o número de bissecções.

Monotônica, se

$$[\mathbf{x}] \subset [\mathbf{y}] \Longrightarrow [\mathbf{f}]([\mathbf{x}]) \subset [\mathbf{f}]([\mathbf{y}]). \tag{5-39}$$

Mínima, se a função de inclusão é a menor caixa que contém $\boldsymbol{f}([\mathbf{x}])$. A função de inclusão mínima é única e representada por $[\mathbf{f}]^*$.

Uma forma de se obter funções de inclusão é pelo método descrito no seguinte teorema (JAULIN *et al.*, 2001)

Teorema 2 *Seja uma função* $f: \mathbb{R}^{n_v} \to \mathbb{R}$ *expressa em termos finitos das operações clássicas e funções elementares (raiz quadrada, exponencial, seno, entre outras). O procedimento de substituir cada ocorrência da variável real pela variável intervalar correspondente e substituir cada operação real pela respectiva operação intervalar gera a função de inclusão natural* $[\mathrm{f}]: \mathbb{IR}^{n_v} \to \mathbb{IR}$. *Toda função de inclusão natural é monotônica e estreita. Se* f *contém somente operadores e funções elementares contínuas, então* $[\mathrm{f}]$ *também é convergente. Se, além disso, cada variável* $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_{n_v}$ *ocorrer no máximo uma vez na expressão de* f, *então* $[\mathrm{f}]$ *também será mínima.*

Dado que nem sempre é possível obter funções de inclusão natural eficientes, outras formas de obter funções de inclusões têm sido propostas, por exemplo, a centralizada, do valor médio, função de inclusão de Taylor, forma do teste de monotocidade, entre outras (MOORE *et al.*, 2009; JAULIN *et al.*, 2001). Mais detalhes podem ser consultados em (NEUMAIER, 2009).

5.3.4 Problema da Dependência

É importante frisar que o resultado da avaliação das funções de inclusão depende de como estas são construídas. Por exemplo, dado $[f]([x]) = [x] * (1 - [x])$ e $[g]([x]) = [x] - [x]^2$. Para $[x] = [0,1]$ tem-se que $[f]([x]) = [0,1]$ e $[g]([x]) = [-1,1]$. Isso indica que essas funções intervalares são distintas, embora na aritmética real as expressões $x - x^2$ e $x * (1 - x)$ sejam equivalentes. Essa situação ocorre porque a propriedade distributiva não é válida na computação intervalar (MOORE *et al.*, 2009). É interessante perceber também que o intervalo resultante de $[g]([x])$ é mais largo quando comparado ao intervalo resultante de $[f]([x])$. Essa é a principal consequência do problema da dependência. Em geral, a causa é atribuída à múltipla ocorrência de uma mesma variável. Uma forma simples de tratar esse problema é por meio da reorganização das expressões matemáticas, visando reduzir a multiplicidade de cada variável. É importante tratar o problema da dependência, pois segundo HANSEN e WALSTER (2004) é desejável que o intervalo resultante seja tão estreito quanto possível.

5.3.5Testes de Inclusão

No contexto deste trabalho, os testes de inclusão exercem uma função importante, pois são utilizados no tratamento das restrições existentes nos problemas de otimização. Antes de definir os testes de inclusão é necessário apresentar o conjunto booleano intervalar, as operações lógicas e os operadores relacionais para intervalos.

O conjunto booleano intervalar é definido por (JAULIN *et al.*, 2001):

$$\mathbb{IB} = \{\emptyset, 0, 1, [0,1]\}, \tag{5-40}$$

sendo $\emptyset$ usado para impossível, 0 para falso, 1 para verdadeiro e $[0,1]$ para indeterminado.

As operações lógicas sobre o conjunto booleano intervalar são definidas por:

$$\begin{aligned} [x] \vee [y] &= \{x \vee y \,|x \in [x], y \in [y]\}, \\ [x] \wedge [y] &= \{x \wedge y \,|x \in [x], y \in [y]\}, \\ \neg [x] &= \{\neg\, x \,|x \in [x]\}, \end{aligned} \tag{5-41}$$

sendo $\vee$, $\wedge$ e $\neg$ respectivamente os operadores lógicos OU, E e negação.

Os operadores relacionais podem ser estendidos para a matemática intervalar da seguinte forma:

$$\begin{array}{ll} ([x] \leq [y]) \to 1 & \text{se } x^+ \leq y^-, \\ ([x] \leq [y]) \to 0 & \text{se } x^- > y^+, \\ ([x] \leq [y]) \to [0,1] & \text{se } (x^+ > y^- \wedge x^- < y^+), \\ ([x] = [y]) \to 1 & \text{se } (x^- = y^- \wedge x^+ = y^+), \text{caso contrário é } 0. \end{array} \tag{5-42}$$

Como exemplo tem-se que:

$$\begin{gathered} ([0,1] \leq [2,3]) \to 1. \\ ([3,4] \leq [0,2]) \to 0. \\ ([2,5] \leq [3,6]) \to [0,1]. \end{gathered}$$

Uma função de teste real é dada por $t: \mathbb{R}^{n_v} \rightarrow \mathbb{B}$. Analogamente, um teste de inclusão, utilizado para provar que todos os pontos de uma caixa satisfazem uma propriedade ou não, é representado por uma função $[t]: \mathbb{IR}^{n_v} \rightarrow \mathbb{IB}$ e é definido por:

$$\begin{aligned} &\forall [\mathbf{x}] \in \mathbb{IR}^{n_v}, ([t]([\mathbf{x}]) = 1) \Longrightarrow (\forall x \in [\mathbf{x}], t(x) = 1), \\ &\forall [\mathbf{x}] \in \mathbb{IR}^{n_v}, ([t]([\mathbf{x}]) = 0) \Longrightarrow (\forall x \in [\mathbf{x}], t(x) = 0). \end{aligned} \tag{5-43}$$

5.4 Subpavimentos

As caixas, apesar de envolverem as funções reais por meio das funções de inclusão, incluindo as funções reais descontínuas, podem conter informações desnecessárias. Isso implica na existência de intervalos superdimensionados, que não trazem informações relevantes, apenas gastam mais recursos computacionais e tornam a função de inclusão imprecisa. De acordo com JAULIN *et al.* (2001), uma forma de obter intervalos mais precisos é com o uso de subpavimentos. Um subpavimento de uma caixa $[\mathbf{x}] \subset \mathbb{R}^{n_v}$ consiste da união de subcaixas não sobrepostas de $[\mathbf{x}]$ com largura diferente de zero e com apenas a fronteira em comum.

Um subpavimento regular **RSP**$[\mathbf{x}]$ de $[\mathbf{x}]$ é computado por finitas bissecções de $[\mathbf{x}]$. O processo de bissecção está descrito no final da Subseção 5.2.1. Um subpavimento regular pode ser representado computacionalmente pela estrutura de dados árvore binária. Essa estrutura facilita o desenvolvimento das quatro principais operações realizadas com subpavimentos: junção de subpavimentos irmãos, teste para verificar se uma caixa está inclusa em um subpavimento, interseção e união de subpavimentos.

Como exemplo, suponha $f(x) = (x^2 + 2x - 1)/x$ e sua respectiva função de inclusão sendo $[f]([x]) = ([x]^2 + 2[x] - 1)/[x]$. Assumindo $[x] = [1, 2]$, tem-se que $[f]([1, 2]) = ([1, 2]^2 + 2[1, 2] - 1)/[1, 2] = [1, 7]$. Porém, tratando $[x] = [1, 2]$ como subpavimento, obtêm-se $[1,\ 1{,}5]$ e $[1{,}5,\ 2]$ após uma bissecção. Dessa forma, $[f]([1{,}2]) = [f]([1,\ 1{,}5]) \cup [f]([1{,}5,\ 2]) = [1{,}3,\ 4{,}7]$. Portanto, o uso de subpavimento conduz a um intervalo de saída mais estreito.

5.5 Conclusão

Neste capítulo apresentou-se um breve histórico da IA e definiram-se os principais conceitos de análise intervalar utilizados neste trabalho. O tratamento de incerteza por meio da IA é interessante por permitir uma computação segura de todo o intervalo considerado. No entanto, ter essa confiabilidade resulta em procedimentos com custo computacional elevado. Sendo assim, é importante a obtenção de boas funções de inclusão e o tratamento adequado para o problema da dependência.

6. REGRESSÃO SIMBÓLICA

O termo regressão é familiar para pesquisadores nas áreas de ciências exatas. A regressão consiste em descobrir coeficientes de uma função pré-determinada que melhor ajusta algumas amostras de dados. A dificuldade na regressão consiste em estabelecer qual função deve ser escolhida para melhor se ajustar às amostras. Essa tarefa requer bastante experiência e engenhosidade do pesquisador. Além disso, até mesmo especialistas têm viés na escolha das funções de ajuste. Por exemplo, alguns trabalham com modelos lineares ou quadráticos, mesmo quando os dados poderiam ter melhor ajuste com o uso de modelos mais complexos (POLI *et al.*, 2008).

Perante o que foi apresentado, pode-se perceber que na regressão ao invés de simplesmente calcular coeficientes numéricos para o modelo previamente estabelecido, a questão chave é decidir que tipo de função melhor se ajusta aos dados. Diante disso, é pertinente considerar a Regressão Simbólica (*Symbolic Regression* - SR), pois essa consiste em encontrar a função que melhor ajusta um conjunto finito de amostras sem fazer nenhum tipo de suposição a respeito da estrutura da função de ajuste (KOZA, 1992).

Nesta tese, SR foi empregada a fim de obter expressões aproximadas para as funções objetivo e restrições de modelos matemáticos relativos a problemas de otimização sujeitos à incerteza. A SR foi utilizada em dois algoritmos baseados em GP propostos e descritos na Seção 7.1. As expressões analíticas retornadas por esses algoritmos possibilitam a obtenção de funções de inclusão para as funções de otimização. Tais funções de inclusão são necessárias para que os algoritmos de otimização robusta com tratamento intervalar das incertezas sejam aplicados.

6.1 Exemplo de Regressão Simbólica

Suponha a existência de uma amostra de valores numéricos (Tabela 6-1) referentes a 30 pontos de uma curva que tem como domínio os números reais no intervalo de $[-1,1]$. Sendo (x_i, y_i) o formato de cada ponto $i = 1, \ldots, 30$, tem-se que x_i é o i-ésimo valor da variável independente x cujo domínio é $[-1,1]$ e y_i é o i-ésimo valor da variável dependente y.

Tabela 6-1 - Amostra de pontos.

i	x_i	y_i	i	x_i	y_i	i	x_i	y_i
1	-0,3339	-0,2472	11	-0,7251	-0,3041	21	-0,2855	-0,2206
2	0,8931	3,0396	12	-0,7883	-0,2705	22	0,6890	1,7165
3	-0,1476	-0,1285	13	-0,4192	-0,2862	23	0,9281	3,3309
4	-0,0083	-0,0082	14	0,3178	0,4612	24	-0,0470	-0,0449
5	0,1427	0,1664	15	0,2386	0,3124	25	0,0649	0,0695
6	-0,2714	-0,2123	16	0,5211	1,0079	26	0,0278	0,0286
7	0,9786	3,7908	17	-0,6696	-0,3204	27	-0,8815	-0,1855
8	0,5791	1,2211	18	0,3483	0,5267	28	0,0030	0,0030
9	-0,9979	-0,0041	19	-0,5607	-0,3237	29	-0,4186	-0,2860
10	-0,1473	-0,1283	20	-0,4732	-0,3051	30	-0,7142	-0,3082

Neste exemplo, objetiva-se encontrar uma função simbólica que melhor aproxima o conjunto de amostras apresentado na Tabela 6-1. Inicialmente, as amostras foram divididas em dois subconjuntos, o primeiro chamado conjunto de treinamento (contendo 24 amostras) e o segundo conjunto de validação (contendo 6 amostras). Cada uma das amostras do conjunto de treinamento é considerada um *fitness case* e é utilizada para nortear o processo de busca. As amostras do conjunto de validação são utilizadas para verificar a qualidade da expressão de ajuste obtida. A definição do conjunto primitivo irá determinar que tipo de expressões poderá ser obtida para tentar encontrar a resposta do problema. Nesta seção, examinam-se dois cenários. No primeiro cenário para compor o conjunto de terminais escolheu-se a variável independente x, portanto tem-se:

$$\mathbf{TS} = \{\mathrm{x}\}. \tag{6-1}$$

No conjunto de funções devem-se utilizar funções e operações aritméticas necessárias para realizar o ajuste dos dados. Supondo que as operações de adição, subtração, multiplicação, divisão protegida (pdiv), seno e cosseno sejam suficientes para o problema, tem-se que:

$$\mathbf{FS} = \{+, -, *, \mathrm{pdiv}, \mathrm{seno}, \mathrm{cosseno}\}. \tag{6-2}$$

A aridade de cada função de **FS** é 2, 2, 2, 2,1 e 1, respectivamente. A divisão protegida consiste em retornar um valor válido para a operação de divisão mesmo que o denominador seja igual a zero. Em geral, por definição, pdiv retorna 1 caso o denominador seja zero.

O *raw fitness* usado em ambos os cenários é o somatório do erro absoluto, ou seja, considerando-se todos os 24 *fitness cases* basta somar os valores absolutos das diferenças entre o valor da variável dependente calculado com a expressão obtida e o valor correto para cada *fitness case*, o qual é fornecido como entrada do problema.

Considerando-se os primeiros 24 pontos da Tabela 6-1 como o conjunto de treinamento, o número máximo de gerações igual a 50, o tamanho da população igual a 500, taxa de reprodução de 0,1, probabilidade de cruzamento de 0,7 e a probabilidade de mutação de 0,3 executou-se um algoritmo de GP. O algoritmo encontrou na décima primeira geração em 10,51 segundos a seguinte expressão para ajustar os dados no primeiro cenário: $\left(((((\mathrm{x} + \mathrm{x} * \mathrm{x}) * \mathrm{x}\right) + \mathrm{pdiv}(\mathrm{x} * \mathrm{x}, \mathrm{x})) * \mathrm{x}) + \mathrm{pdiv}(\mathrm{x} * \mathrm{x}, \mathrm{x}))$, cujo *raw fitness* é $2{,}7662 * 10^{-14}$.

Para verificar a qualidade da expressão obtida comparou-se o resultado da expressão de ajuste para os seis pontos do conjunto de validação (pontos $i = 25, \ldots, 30$) com os valores corretos de saída para esses pontos. Na comparação obteve-se *raw fitness* (erro absoluto) de $8{,}6172 * 10^{-16}$ em relação ao conjunto de validação. A Figura 6-1 A exibe os resultados encontrados, os quais confirmam a boa qualidade do ajuste.

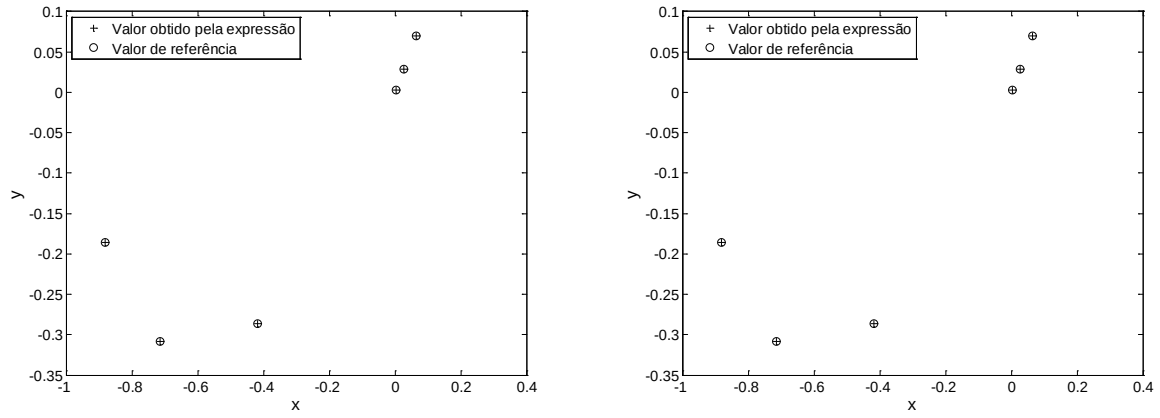


Figura 6-1-Pontos de referência e pontos encontrados pela expressão de ajuste no: A) cenário 1 e B) cenário 2.

A partir desse resultado pode ser feita a seguinte pergunta: será que existem outras expressões que também ajustam esses mesmos pontos com boa qualidade? Para responder essa questão, pode-se alterar o conjunto primitivo de modo a introduzir novas funções e/ou terminais buscando obter diferentes formatos para a expressão de ajuste. Suponha um cenário 2 no qual os conjuntos de terminais e funções sejam dados, respectivamente, por:

$$\mathbf{TS} = \{x, I\} \text{ e } \mathbf{FS} = \{+, ^\wedge\}, \tag{6-3}$$

sendo I uma constante efêmera inteira, aleatória, pertencente ao conjunto {1,2,3,4,5,6,7,8}. Cada I existente nas expressões, no momento de avaliação das mesmas, é substituído por uma constante escolhida aleatoriamente pertencente ao conjunto especificado. Caso a expressão que contenha a(s) constante(s) tenha um bom valor de *raw fitness* tem-se que, muito provavelmente, essa expressão e consequentemente a(s) constante(s) seguirão nas gerações futuras, caso contrário, provavelmente será descartada. Por esse motivo a constante é denominada efêmera.

Considerando-se os primeiros 24 pontos da Tabela 6-1 como o conjunto de treinamento, o número máximo de gerações igual a 50, o tamanho da população igual a 500, taxa de reprodução de 0,1, probabilidade de cruzamento de 0,7 e a probabilidade de mutação de 0,3 executou-se um algoritmo de GP. O algoritmo encontrou na vigésima quarta geração em 25,48 segundos a seguinte expressão para ajustar os dados no cenário 2: $x^{2^2} + x + x^3 + x^2$, cujo *raw fitness* é $2{,}794*10^{-14}$. Ressalta-se que as constantes inteiras 2 e 3 presentes como expoente na expressão foram geradas devido ao símbolo I existente no conjunto de terminais. Para avaliar a qualidade do ajuste obtido nesse caso, apresenta-se a Figura 6-1 B que mostra a comparação dos pontos do conjunto de validação com os de treinamento utilizando-se a expressão obtida. O *raw fitness* (erro absoluto) obtido nesse caso foi de $9{,}1723 * 10^{-16}$ em relação ao conjunto de validação.

6.2 Análise Intervalar na Regressão Simbólica

No exemplo da seção anterior utilizou-se no conjunto de funções um operador protegido. No entanto, KEIJZER (2003) mostra que apesar dos operadores protegidos evitarem um comportamento matemático indefinido das expressões geradas, há casos em que essa técnica possui deficiências, principalmente na vizinhança das singularidades matemáticas. HOWARD e ROBERTS (2001) identificaram falhas com a divisão protegida, pois no problema tratado por estes, os valores arbitrários usados por esse operador induziam de forma rápida a convergência para um ótimo local que usava esses valores. Diante disso, necessita-se contornar essa questão. Uma possibilidade, que não utiliza operadores protegidos, é excluir as expressões que gerem alguma inconsistência matemática. Essa técnica pode funcionar bem para o conjunto treinamento. Entretanto, não há nenhuma garantia quanto ao conjunto de validação e demais casos não treinados.

KEIJZER (2003) propõe usar a matemática intervalar para verificar se para todo o intervalo das variáveis de entrada é possível ter uma saída definida, sendo assim a expressão é considerada válida, caso contrário, a expressão é invalidada. Somente as expressões válidas são submetidas ao processo de avaliação. Dessa forma, pode-se garantir que tanto para o conjunto de treinamento quanto para os demais casos não treinados a expressão terá o mesmo comportamento. Os resultados do trabalho mostram que essa técnica contorna completamente a necessidade de proteger os operadores matemáticos, e que é menos suscetível à ocorrência de *overfitting* (geralmente ocorre quando no ajuste dos dados o erro no conjunto de treinamento decresce enquanto no conjunto de validação o erro cresce).

6.3 Regressão Simbólica com *Linear Scaling*

De acordo com KEIJZER (2003), em problemas de regressão simbólica é comum usar o erro absoluto ou o erro quadrático como medida de *fitness*, o que leva a uma comparação direta entre os valores de referência e os obtidos de cada indivíduo. Com isso, o algoritmo de GP necessita primeiro encontrar a faixa correta dos valores antes de considerar as soluções, o que é uma tarefa dispendiosa. Para verificar isso, KEIJZER (2003) comparou os resultados da aplicação de um algoritmo de GP entre a função polinomial (x^2) e uma variação dessa, a função $x^2 + 100$. A análise dos resultados mostrou que a segunda função tem maior dificuldade em convergir corretamente, justamente pela mudança de escala devido à adição da constante. Para evitar a preocupação com a escala das funções e fazer com que o algoritmo de GP concentre-se em buscar a expressão que retorne a forma desejada da saída dos dados independentemente da escala, KEIJZER (2004) propõe o uso de regressão linear. Esse procedimento consiste em calcular duas constantes, o intercepto (a) e a inclinação (b) para cada expressão gerada pela GP. Sendo y as saídas da expressão para os n *fitness cases* e r os valores de referência desses tem-se que:

$$b = \frac{\sum_{i=1}^{n}[(r_i - \bar{r})(y_i - \bar{y})]}{\sum_{i=1}^{n}[(y_i - \bar{y})^2]}$$
$$a = \bar{r} - b\bar{y}, \tag{6-4}$$

sendo $\bar{r}$ e $\bar{y}$ a média de r e a média de y, respectivamente. Ressalta-se que a e b são obtidos de modo que a soma dos quadrados do erro entre os valores de referência (r) e $(a + by)$ seja minimizada.

Uma vez encontrados os valores de a e b de uma expressão, a medida de erro dessa expressão em relação aos valores de referência pode ser calculada pelo erro quadrático médio (*Mean Squared Error* - MSE) como segue:

$$\frac{1}{n}\sum_{i=1}^{n}(a + by_i - r_i)^2 \tag{6-5}$$

KEIJZER (2003) ressalta que a variância de y é medida a fim de assegurar a estabilidade numérica no cálculo da inclinação e do intercepto. As expressões cujo valor da variância esteja fora do intervalo $[10^{-7}, 10^{7}]$ são excluídas.

Devido ao uso da regressão linear, o algoritmo de GP foca o seu esforço em encontrar um formato adequado para ajustar os valores de referência, sem se preocupar com a escala adequada. A prova do cálculo de a e b é encontrada em (KEIJZER, 2004) e diversos experimentos usando *linear scaling* são apresentados em (KEIJZER, 2003).

6.4 Conclusão

Inicialmente, neste capítulo, introduziu-se o conceito de SR e apresentou-se um exemplo de funcionamento. Depois, duas melhorias propostas na literatura para lidar com SR foram descritas, são elas: o uso de análise intervalar e de *linear scaling* na regressão simbólica. Observa-se que uma diferença da SR quando comparada a outras formas de regressão é o fato de não precisar estabelecer a estrutura da função de ajuste, pois a construção desta função é parte do processo de regressão.

7. ALGORITMOS E CONCEITOS PROPOSTOS

7.1 Aproximação de Funções de Inclusão Via Programação Genética

Nesta seção apresentam-se dois algoritmos propostos, um mono-objetivo chamado SNIF-GPA (*Surrogate Natural Inclusion Function by Genetic Programming Algorithm*) e, outro multiobjetivo, denominado SNIF-MOGPA (*Surrogate Natural Inclusion Function by Multi-Objective Genetic Programming Algorithm*). Inicialmente, os aspectos gerais e comuns dos algoritmos são expostos e, em seguida, ambos são detalhados. A importância dos algoritmos propostos é o fato deles serem capazes de gerar as funções de inclusão necessárias para os algoritmos de otimização robusta baseados em análise intervalar. A geração das funções de inclusão pelos algoritmos baseia-se na seguinte ideia: primeiramente, o algoritmo (SNIF-GPA ou SNIF-MOPGA) obtém uma expressão analítica aproximada de acordo com a entrada fornecida, em seguida, aplica-se o Teorema 2 (descrito na Subseção 5.3.3) na expressão obtida e, como resultado, tem-se a função de inclusão natural aproximada.

7.1.1 Estrutura dos Indivíduos no SNIF-GPA e SNIF-MOGPA

Em GP os indivíduos costumam ser representados pela estrutura de dados em árvore, a qual deve ser genérica para não limitar o processo de busca. O uso de árvore binária restringiria o algoritmo a trabalhar com funções cujo valor da aridade seria de no máximo 2. Assim, optou-se por utilizar a árvore n-ária nos algoritmos, pois essa é capaz de oferecer maior generalidade. Na representação proposta, cada nó da árvore possui a estrutura apresentada na Figura 7-1.

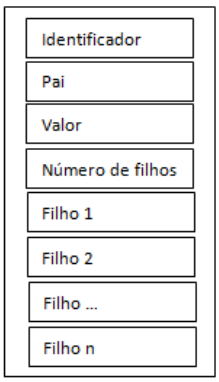


Figura 7-1 – Estrutura geral do nó da árvore.

O campo identificador representa a chave do nó. Esse campo deve ser distinto para cada um dos nós. O valor do identificador é definido automaticamente pelo algoritmo considerando-se a ordem de criação dos nós, isto quer dizer que o primeiro nó criado possui valor 1 para o identificador, o segundo nó criado possui valor 2 para o identificador e assim sucessivamente. O campo pai é o valor da chave do nó-pai do referido nó. Para o nó-raiz o valor do campo pai, por convenção, é definido

como zero. O campo valor possui qualquer valor do conjunto primitivo, ou seja, uma função ou terminal. O campo número de filhos indica quantos nós-filho o nó possui, ou seja, se o valor for uma função será a aridade dessa, se o valor for um terminal convencionou-se o valor -1 para esse campo. Os campos filho 1, filho 2, filho n são criados dinamicamente de acordo com o campo número de filhos. O valor contido nesses campos indica o valor do identificador do n-ésimo filho deste nó.

Uma árvore é formada por um ou mais nós. De acordo com a definição do valor do identificador de cada nó, explicada no parágrafo anterior, pode-se concluir que o nó raiz sempre terá o identificador igual a 1. A seguir, como exemplo, a Figura 7-2 representa a estrutura de nós para a expressão $\sin(x) + x + x$:

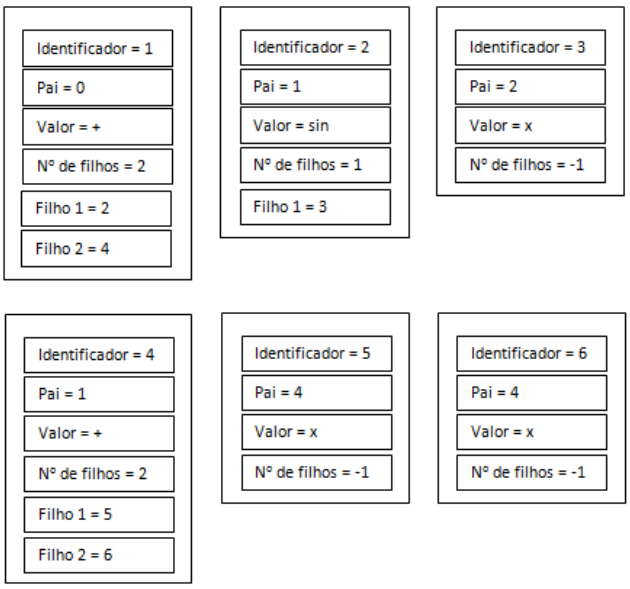


Figura 7-2 – Estrutura de nós para a expressão $\sin(x) + x + x$.

7.1.2 Estrutura Geral do SNIF-GPA

O Algoritmo 7-1 descreve as etapas da fase de treinamento do SNIF-GPA. Na etapa 1 gera-se a população inicial de programas. Há três formas implementadas para esse procedimento: *full*, *grow* e *ramped half-and-half*. Na etapa 2 é possível gerar os *fitness cases* aleatoriamente dentro do domínio de cada umas das variáveis de decisão. Outra opção disponível é realizar a leitura dos dados a serem aproximados a partir de um arquivo texto. Na etapa 3 cada programa tem sua estrutura de árvore convertida em uma expressão matemática. Esse procedimento é explicado em detalhes adiante na Subseção 7.1.2.1. Na etapa 5.1 os programas são selecionados para as operações genéticas. Na etapa 5.2 o cruzamento e a mutação são realizados conforme descrito na Subseção 4.4.3. Para eliminar programas repetidos (etapa 5.5) utilizou-se um critério de similaridade, o qual considera que um programa é similar ao outro se esses possuem mesmo valor de *fitness*, mesmo número de nós terminais e mesma quantidade de nodos não terminais. Uma vez que há possibilidade de eliminação de programas, a população pode variar de tamanho ao longo das gerações, naturalmente, até o limite permitido. Na etapa 5.7 é garantido que o tamanho máximo da população será respeitado.

Algoritmo 7-1– Fase de treinamento do SNIF-GPA.

1. Crie aleatoriamente uma população inicial (P_p) com n programas de acordo com o conjunto primitivo adequado ao domínio do problema.
2. Determine os *fitness cases* para o problema.
3. Execute cada programa e determine seu valor de *fitness.*
4. Faça o *ranking* dos programas da população em relação ao valor de *fitness*.
5. Até que uma solução aceitável seja encontrada ou algum critério de parada seja satisfeito, repita:
   5.1. Execute o processo de seleção em P_p.
   5.2. Realize os processos de cruzamento e mutação gerando a população filha (P_f).
   5.3. Execute cada programa da população filha e avalie o seu valor de *fitness.*
   5.4. Junte as duas populações P_p e P_f em P_t.
   5.5. Elimine programas similares.
   5.6. Faça o *ranking* dos programas de acordo com o valor de *fitness*.
   5.7. Selecione para a população seguinte os n primeiros programas de P_t e os armazene em P_p.
6. Retorne o melhor programa encontrado.

O Algoritmo 7-1 requer a configuração de alguns parâmetros de entrada. Os itens que devem ser definidos são: o número de programas da população; o número máximo de gerações; as probabilidades de reprodução, cruzamento e mutação; o tamanho do torneio; o problema a ser tratado; o modo de entrada de dados para ajuste; o número máximo de iterações sem melhora do *fitness*; a forma de geração da população inicial; a altura máxima da árvore na geração inicial; a altura máxima da árvore após o cruzamento; precisão para que um programa seja considerado como melhor solução do problema; o conjunto de funções, o conjunto de terminais e o intervalo de geração das constantes.

7.1.2.1 Função de Avaliação

Para atribuir um valor de aptidão a cada programa calcula-se o *raw fitness*, dado pelo erro quadrático médio com *linear scaling*, conforme descrito na Seção 6.3. Somente é calculada a *fitness* dos programas que gerem resultados válidos, o que é verificado pela aplicação da análise intervalar conforme descrito na Seção 6.2. Um procedimento que é essencial nessa etapa é a conversão da estrutura de árvore em uma expressão matemática válida. Essa conversão para problemas cujas funções possuem aridade de no máximo 2 é realizada conforme o Algoritmo 7-2. Devido à possibilidade de existência da constante efêmera aleatória no processo de conversão, faz-se necessário atualizar a estrutura da árvore com a constante gerada.

Algoritmo 7-2– Conversão da árvore em uma expressão.

1. Enquanto houver nó na árvore
   1.1. Se o valor do nó é um operador binário
      1.1.1. Chame essa função recursivamente para subárvore esquerda.
      1.1.2. Imprima valor do nó.
      1.1.3. Chame essa função recursivamente para subárvore direita.
   1.2. Senão se valor do nó é operador unário
      1.2.1. Imprima valor do nó.
      1.2.2. Chame essa função recursivamente para subárvore esquerda.
   1.3. Senão *// só pode ser terminal*
      1.3.1. Se valor do nó for I
         1.3.1.1. Gere um valor para a constante efêmera aleatória.
      1.3.2. Fim se
      1.3.3. Imprima valor do nó.
   1.4. Fim se
2. Fim Enquanto

7.1.2.2 Cruzamento e Mutação

Em termos de implementação, o ponto crítico do cruzamento refere-se ao tratamento do *bloat* e à atualização da chave (identificador) de cada um dos nós das árvores envolvidas na operação. O *bloat* ocorre quando o tamanho da expressão cresce sem que haja melhora significativa no valor de aptidão. O *bloat* conduz o algoritmo a produzir expressões mais complexas no que diz respeito à interpretação. O trabalho de OLIVEIRA DA SILVA (2008) descreve detalhadamente esse problema, bem como diversas propostas de tratamento do mesmo. Nos algoritmos de GP propostos neste trabalho, o tratamento do *bloat* consistiu na definição de uma altura máxima para a árvore, dessa forma, as árvores que extrapolam esse limite são descartadas. Além disso, os algoritmos foram implementados para escolher os nós de cruzamento em ambas as árvores de modo que a diferença entre a altura desses seja de no máximo um. Isso evita o risco de *bloat* (VLADISLAVLEVA *et al.*, 2009). No entanto, a obtenção dessa diferença de altura de no máximo um nem sempre é possível.

No que tange ao controle das chaves dos nós das árvores a questão é a seguinte: ao realizar o cruzamento entre dois indivíduos, faz-se necessário permutar as subárvores cujas raízes sejam os pontos de cortes sorteados, de modo que não se perca a referência de nenhum nó e, além disso, que cada nó possua uma chave única na árvore em que está. Para garantir que não haja chaves repetidas constroem-se as árvores filhas a partir das árvores pais, respeitando-se o ponto de corte selecionado em cada uma. Com isso, a árvore filha 1 é construída pegando-se todos os elementos da árvore pai 1 até o nó que antecede o ponto de corte dessa. Feito isso, pegam-se todos os elementos da subárvore 2 (extraída da árvore pai 2) e por fim pegam-se os elementos da árvore pai 1 após o ponto de corte 1 e que não façam parte da subárvore 1 (extraída da árvore pai 1). A árvore filha 2 é construída de forma análoga.

No procedimento de mutação, o único detalhe é que o próximo procedimento após a mutação deve ser a conversão da estrutura de árvore em expressão, pois na mutação pode acontecer de um terminal ser mutado para um símbolo de constante efêmera aleatória. Nesse caso, esse símbolo deve ser convertido em uma constante válida antes de qualquer outro procedimento ser realizado com esse indivíduo.

7.1.2.3 Fase de Validação do SNIF-GPA

O Algoritmo 7-3 é usado na fase de validação do SNIF-GPA e consiste em retornar a expressão com a menor média da tolerância considerando-se todos os grupos de validação. O conceito de tolerância no contexto do SNIF-GPA é apresentado adiante, após a definição do erro percentual.

Algoritmo 7-3– Fase de validação do SNIF-GPA.

1. Defina os j grupos de validação.
2. Para cada expressão k obtida na fase de treinamento faça:
   2.1. Para cada grupo j faça:
      2.1.1. Para cada amostra m do grupo j faça:
         2.1.1.1. Avalie a expressão k tendo como argumento de entrada a amostra m.
         2.1.1.2. Calcule o erro percentual (δ) da expressão k com argumento m.
      2.1.2. Calcule a tolerância da expressão k em relação ao grupo j.
   2.2. Calcule a média da tolerância da expressão k considerando todos os grupos.
3. Retorne a expressão com menor média da tolerância.

No passo 2.1.1.2, o erro percentual para cada amostra é dado pela Equação (7-1):

$$\delta_i = \frac{|r_i - y_i|}{|r_i|} * 100\,, \forall i = 1, \dots, m \text{ e } r_i \neq 0. \tag{7-1}$$

sendo r o valor de referência e y o valor retornado pela expressão gerada pelo algoritmo na fase de treinamento. No passo 2.1.2, a tolerância da expressão relacionada a um determinado grupo de validação é definida como o valor do erro percentual do 51º percentil considerando-se as m amostras do j-ésimo grupo de validação. Em outras palavras, supondo cem amostras, a tolerância da expressão consiste no valor do erro percentual associado à quinquagésima primeira amostra (dado que as amostras estão em ordem ascendente, sendo a chave de ordenação o erro percentual de cada amostra). Para determinar a média da tolerância de uma expressão, basta realizar o somatório da tolerância associada a cada um dos grupos de validação e dividir pela quantidade de grupos do conjunto de validação.

7.1.3 Estrutura Geral do SNIF-MOGPA

Como visto anteriormente, é comum em GP o crescimento descontrolado das expressões durante o processo evolutivo. Quando este aumento não está acompanhado de melhora no valor de *fitness* ocorre o *bloat*. Sabendo-se que na prática os modelos menores são mais fáceis de serem compreendidos, é importante ter estratégias para controle do *bloat* (POLI *et al.*, 2008). Conforme SMITS e KOTANCHEK (2004) uma abordagem é considerar o desempenho e a complexidade das expressões simultaneamente na função de *fitness*. O desempenho define a qualidade da expressão e caracteriza o quão bem essa se ajusta aos dados de treinamento. A complexidade da expressão está relacionada com sua estrutura. Sendo assim, pode-se formular o problema de regressão simbólica em uma versão multiobjetivo como sendo:

$$\begin{aligned} &\text{Minimize} && f_1(\text{exp}), f_2(\text{exp}) \\ &\text{Sujeito a:} && h_1(\text{exp}) = 0 \\ & && \text{exp} \in \mathbf{S}\,, \end{aligned} \tag{7-2}$$

sendo os objetivos f_1 e f_2 o desempenho e a complexidade de exp, respectivamente, e exp uma expressão composta de quaisquer combinações entre funções e terminais. A restrição h_1 é igual a zero se cada variável de decisão presente no conjunto de terminais aparece ao menos uma vez em exp. **S** é o conjunto de todas as possíveis expressões obtidas a partir do conjunto primitivo. Note que na formulação (7-2) a restrição é opcional e depende das necessidades do problema.

Para mensurar o desempenho no SNIF-MOGPA utilizou-se o erro quadrático médio com *fitness scaling*, conforme descrito na Seção 6.3. Para o cálculo de complexidade empregaram-se a altura da árvore e o somatório do número de nós de todas as subárvores. De acordo com SMITS e KOTANCHEK (2004), o somatório do número de nós de todas as subárvores favorece as estruturas balanceadas quando o número de nós é equivalente (veja a Figura 7-3). Além disso, como mostrado na Figura 7-4, essa medida retorna valores diferentes nos casos em que distintos genótipos resultam no mesmo fenótipo.

Figura 7-3 – Ambas as árvores têm três nós. Entretanto, o somatório do número de nós de todas as subárvores é diferente: para a árvore A é 5 = 3 + 1 + 1 e para a árvore B é 6 = 3 + 2 + 1.

Figura 7-4 – Apesar das duas árvores serem equivalentes a $x^2 + 4x + 4$, a da esquerda tem complexidade de 11 = 5+3+1+1+1, a da direita tem complexidade de 37 = 11+9+3+1+1+5+3+1+1+1+1. Neste caso, o vetor objetivo da primeira expressão domina o da segunda.

O algoritmo multiobjetivo de GP proposto nesta tese (Algoritmo 7-4 – fase de treinamento) possui algumas diferenças em relação à versão mono-objetivo apresentada anteriormente. Os pontos distintos são detalhados a seguir: 1) No passo 4, o procedimento usando no *ranking* é o *dominance depth* (o mesmo usado no NSGA-II), o qual ordena a população em diferentes níveis de não dominância. Como as soluções em um mesmo nível são incomparáveis, utiliza-se a distância de multidão como segundo critério de ordenação. 2) Nos passos 3 e 5.3 o valor de *fitness* não é calculado para expressões que não satisfazem a restrição. A essas expressões é atribuído um valor ruim (alto) de *fitness*. 3) No passo 5.5 para avaliar a similaridade, como desta vez têm-se mais de uma função objetivo, deve-se considerar cada um dos valores de *fitness*, ou seja, o valor para o desempenho e o valor para complexidade.

Algoritmo 7-4– Fase de treinamento do SNIF-MOGPA.

1. Crie aleatoriamente uma população inicial (P_p) com n programas de acordo com o conjunto primitivo adequado ao domínio do problema.
2. Determine os *fitness cases* para o problema.
3. Execute cada programa e determine seu valor de *fitness.*
4. Faça o *ranking* de acordo com a dominância e a distância da multidão.
5. Até que uma solução aceitável seja encontrada ou algum critério de parada seja satisfeito, repita:
   5.1. Execute o processo de seleção em P_p.
   5.2. Execute os processos de cruzamento e mutação gerando a população filha (P_f).
   5.3. Avalie cada programa da população filha e determine seu valor de *fitness.*
   5.4. Junte as duas populações P_p e P_f em P_t.
   5.5. Elimine programas similares.
   5.6. Faça o *ranking* dos programas de acordo com a dominância e a distância da multidão.
   5.7. Selecione para a população seguinte os n primeiros programas de P_t e os armazene em P_p.
6. Retorne os melhores programas não dominados encontrados.

7.1.3.1 Fase de Validação do SNIF-MOGPA

O Algoritmo 7-5, apresentado nesta seção, calcula a tolerância nos grupos de validação para todas as expressões obtidas na fase de treinamento do SNIF-MOGPA. A distinção em relação ao Algoritmo 7-3 descrito previamente consiste nos passos 3 e 4. No passo 3 utiliza-se o *dominance depth* para realizar o *ranking* das expressões, no passo 4 retornam-se somente as expressões não-dominadas.

Algoritmo 7-5– Fase de validação do SNIF-MOGPA.

1. Defina os j grupos de validação.
2. Para cada expressão k obtida na fase de treinamento faça:
   2.1. Para cada grupo j faça:
      2.1.1. Para cada amostra m do grupo j faça:
         2.1.1.1. Avalie a expressão k tendo como argumento de entrada a amostra m.
         2.1.1.2. Calcule o erro percentual (δ) da expressão k com argumento m.
      2.1.2. Calcule a tolerância da expressão k em relação ao grupo j.
   2.2. Calcule a média da tolerância da expressão k considerando todos os grupos.
3. Faça o *ranking* das k expressões usando a média da tolerância e a medida de complexidade com base em critérios de não-dominância.
4. Retorne as expressões não-dominadas.

7.2 Maximização da Influência das Incertezas no Contexto Intervalar

Nesta seção descreve-se uma forma de maximizar a influência das incertezas (ponto ideal de maximização - $\mathbf{u}^{max}$) a fim de encontrar o pior caso de atuação dessas sob cada solução viável (SOARES, 2008). Além disso, propõe-se uma forma alternativa (fronteira ideal de maximização - $\mathbf{F}^{max}$) de realizar esse processo de maximização da influência das incertezas. Esses conceitos são importantes em otimização robusta, pois irão nortear todo o processo de busca das soluções.

Conforme mostrado no capítulo 2 a otimização multiobjetivo baseia-se na comparação de dominância entre vetores reais. Como nesta tese utiliza-se a IA na computação da incerteza faz-se necessário estender o conceito de dominância para caixas intervalares, o que é mostrado a seguir na Definição 7-1 baseada em (SOARES, 2008).

Definição 7-1 *Dado* $\mathbf{u} \in \mathbb{R}^{n_f}$, $[\mathbf{k}] = [\mathbf{k}^-, \mathbf{k}^+] \in \mathbb{IR}^{n_f}$ e $[\mathbf{n}] = [\mathbf{n}^-, \mathbf{n}^+] \in \mathbb{IR}^{n_f}$ *estende-se o operador* $\prec$ *para o uso com caixas intervalares como segue*

$$\mathbf{u} \prec [\mathbf{k}] \Leftrightarrow \mathbf{u} \prec \mathbf{k}^-; \tag{7-3}$$

$$[\mathbf{n}] \prec [\mathbf{k}] \Leftrightarrow \mathbf{n}^+ \prec \mathbf{k}^-. \tag{7-4}$$

Em otimização multiobjetivo robusta, para calcular a ação do conjunto de incertezas $\mathbf{P} \subseteq \mathbb{R}^{n_p}$ sob cada solução $\mathbf{x} \in \mathbf{S}$ deve-se considerar a influência de cada $\mathbf{p} \in \mathbf{P}$ sobre cada $\mathbf{x}$. Do ponto de vista prático, para realizar isso via tratamento intervalar das incertezas, necessita-se gerar um subpavimento para **P**, isto é, representar **P** por um conjunto de caixas intervalares não sobrepostas. O Algoritmo 7-6 descreve o procedimento recursivo para a geração do subpavimento do espaço de incertezas **P**. Ressalta-se que o Algoritmo 7-6 deve ser chamado com o parâmetro $[\mathbf{p}]$ sendo a caixa que contém todo o espaço de incertezas **P** e o parâmetro subpavimento sendo $\emptyset$. O parâmetro ε_p indica o nível de precisão desejado no subpavimento. Como retorno do algoritmo obtém-se o subpavimento de **P**.

Algoritmo 7-6– Geração do subpavimento para o espaço de incertezas.

1. funcao subpavimento = gera_subpavimento ($[\mathbf{p}]$, ε_p, subpavimento)
2. Se (w($[\mathbf{p}]$) < ε_p) *// se a largura (Definição 5-9) de* $[\mathbf{p}]$ *satisfaz a precisão* ε_p.
   2.1. Coloque $[\mathbf{p}]$ no subpavimento. *// a caixa* $[\mathbf{p}]$ *é inserida no subpavimento.*
   2.2. Retorne. *// finaliza função gera_subpavimento.*
3. Fim Se
4. Bissecte $[\mathbf{p}]$ e obtenha $[\mathbf{p}]_1$ e $[\mathbf{p}]_2$. *// A bissecção de* $[\mathbf{p}]$ *é feita conforme a Definição 5-12.*
5. subpavimento = gera_subpavimento ($[\mathbf{p}]_1$, ε_p, subpavimento) *// chama a função recursivamente para* $[\mathbf{p}]_1$
6. subpavimento = gera_subpavimento ($[\mathbf{p}]_2$, ε_p, subpavimento) // *chama a função recursivamente para* $[\mathbf{p}]_2$
7. fim funcao

Uma vez obtido o subpavimento do espaço das incertezas, computar $[\mathbf{f}](\mathbf{x}, \mathbf{P})$ para um dado $\mathbf{x} \in \mathbf{S}$ significa calcular $[\mathbf{f}](\mathbf{x}, [\mathbf{p}]_1)$, $[\mathbf{f}](\mathbf{x}, [\mathbf{p}]_2)$, ... , $[\mathbf{f}](\mathbf{x}, [\mathbf{p}]_{n_s})$, sendo $n_s =$ número de caixas intervalares que compõem o subpavimento do espaço de incertezas. Por exemplo, como mostrado na Figura 7-5, suponha que **P** seja representado pelo subpavimento $\{[\mathbf{p}]_1, [\mathbf{p}]_2, [\mathbf{p}]_3, [\mathbf{p}]_4\}$, com isso computar $[\mathbf{f}](\mathbf{x}, \mathbf{P})$ implica em calcular $[\mathbf{f}](\mathbf{x}, [\mathbf{p}]_1)$, $[\mathbf{f}](\mathbf{x}, [\mathbf{p}]_2)$, $[\mathbf{f}](\mathbf{x}, [\mathbf{p}]_3)$ e $[\mathbf{f}](\mathbf{x}, [\mathbf{p}]_4)$. Neste caso, como retorno obtêm-se quatro caixas intervalares, uma para cada caixa do subpavimento de **P**.

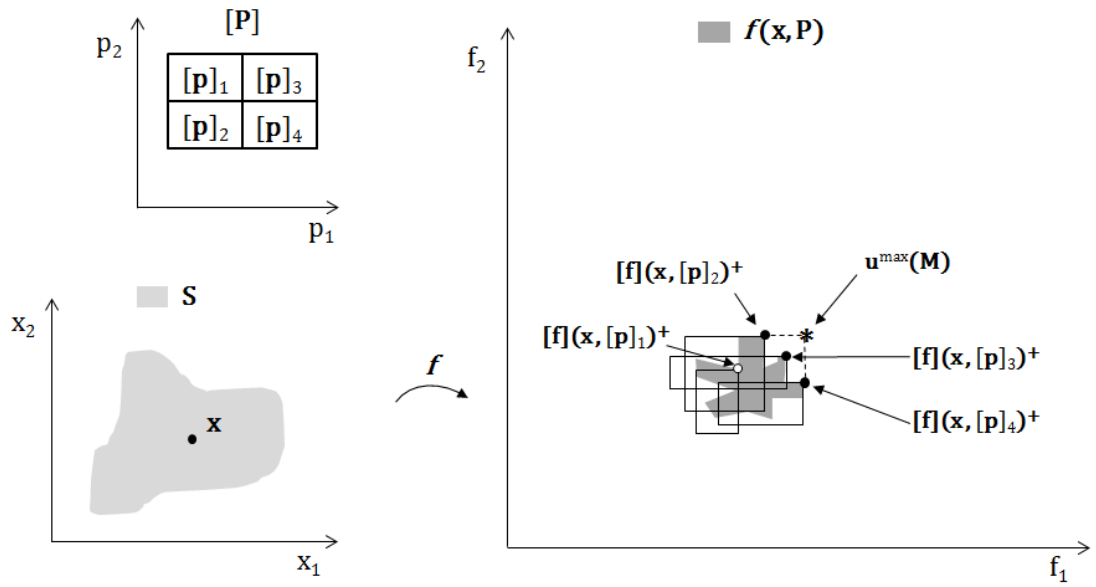


Figura 7-5-Cálculo de **M**, $\mathbf{u}^{max}$ e $\mathbf{F}^{max}$ de um ponto viável **x** sujeito às incertezas em **P**.

Dada uma caixa $[\mathbf{f}](\mathbf{x},[\mathbf{p}]_i)$ no espaço dos objetivos, para determinar o ponto em que ocorre a máxima (pior caso) influência das incertezas na solução viável **x** considerando-se a i-ésima caixa do subpavimento de **P**, deve-se pegar o limite superior da caixa no espaço dos objetivos que representa a computação das funções objetivo dada a atuação da i-ésima caixa de incerteza sobre **x**, que é o ponto $[\mathbf{f}](\mathbf{x},[\mathbf{p}]_i)^+$ (Definição 5-8). Assumindo $i = 1,\dots,n_s$, define-se $[\mathbf{f}](\mathbf{x},\mathbf{P})^+$ por:

$$\mathbf{M} = [\mathbf{f}](\mathbf{x},\mathbf{P})^+ = \left\{[\mathbf{f}](\mathbf{x},[\mathbf{p}]_1)^+,[\mathbf{f}](\mathbf{x},[\mathbf{p}]_2)^+,\dots,[\mathbf{f}]\left(\mathbf{x},[\mathbf{p}]_{n_s}\right)^+\right\}. \tag{7-5}$$

Note que o número de pontos em **M**, que é igual ao valor de n_s, depende do valor do parâmetro ε_p do Algoritmo 7-6. No exemplo da Figura 7-5, o conjunto **M** é constituído de quatro pontos (bolinha branca e bolinhas pretas) no espaço dos objetivos.

Diante disso, SOARES (2008) define o conceito de ponto ideal de maximização $\mathbf{u}^{max}$, o qual ao ser aplicado sobre **M** resulta em um único ponto, se $\mathbf{M} \neq \emptyset$. Porém, SOARES (2008) ressalta que $\mathbf{u}^{max}(\mathbf{M})$ pode não pertencer a $\boldsymbol{f}(\mathbf{x},\mathbf{P})$ e, por consequência não pertencer à imagem viável. No exemplo da Figura 7-5, $\mathbf{u}^{max}(\mathbf{M})$ é representado pelo asterisco preto.

Definição 7-2 *Considere o conjunto de vetores* $\mathbf{U} = \{\mathbf{a},\mathbf{b},\dots,\mathbf{y}\} \subset \mathbb{R}^{n_f}$. *O ponto ideal de maximização* $\mathbf{u}^{max} \in \mathbb{R}^{n_f}$ *do conjunto* **U** *é*

$$u_k^{max}(\mathbf{U}) = \max_k \{a_k, b_k, \dots, y_k\}, k = 1,\dots,n_f. \tag{7-6}$$

Nesta tese, propõe-se o conceito de fronteira ideal de maximização $\mathbf{F}^{max}$, o qual é apresentado a seguir.

Definição 7-3 *Considere o conjunto de vetores* $\mathbf{U} = \{\mathbf{u}_1,\mathbf{u}_2,\dots,\mathbf{u}_{n_s}\}, \mathbf{U} \subseteq \mathbb{R}^{n_f}$. *A fronteira ideal de maximização* $\mathbf{F}^{max}$ *do conjunto* **U** *é*

$$\mathbf{F}^{\max}(\mathbf{U}) = \{\mathbf{u} \in \mathbf{U} \mid \nexists\, \mathbf{u}' \in \mathbf{U}, \mathbf{u} \prec \mathbf{u}'\}. \tag{7-7}$$

No exemplo da Figura 7-5, $\mathbf{F}^{\max}(\mathbf{M}) = \{[\mathbf{f}](\mathbf{x}, [\mathbf{p}]_{\mathbf{2}})^{+}, [\mathbf{f}](\mathbf{x}, [\mathbf{p}]_{\mathbf{3}})^{+}, [\mathbf{f}](\mathbf{x}, [\mathbf{p}]_{\mathbf{4}})^{+}\}$, representado pelas bolinhas pretas.

Uma vantagem de $\mathbf{F}^{\max}$ é que se $[\mathbf{f}](\mathbf{x}, [\mathbf{p}]_{\mathbf{i}})^{+}$ pertencer a $\boldsymbol{f}(\mathbf{x}, [\mathbf{p}]_{\mathbf{i}})$ implica que $\mathbf{F}^{\max}(\mathbf{M})$ resultará em pontos na imagem viável, sendo que i representa o índice das caixas do subpavimento **P** referente aos pontos que fazem parte de $\mathbf{F}^{\max}(\mathbf{M})$. Porém, podem existir casos em que $[\mathbf{f}](\mathbf{x}, [\mathbf{p}]_{\mathbf{i}})^{+}$ não pertence a $\boldsymbol{f}(\mathbf{x}, [\mathbf{p}]_{\mathbf{i}})$, uma vez que $[\mathbf{f}](\mathbf{x}, [\mathbf{p}]_{\mathbf{i}})$ é uma função de inclusão para $\boldsymbol{f}(\mathbf{x}, [\mathbf{p}]_{\mathbf{i}})$.

Outra característica de $\mathbf{F}^{\max}$ é que pela própria definição de $\mathbf{u}^{\max}(\mathbf{M})$, dado **M**, com $\mathbf{M} \neq \emptyset$, todos os pontos de $\mathbf{F}^{\max}(\mathbf{M})$ dominam $\mathbf{u}^{\max}(\mathbf{M})$, exceto no caso em que $\mathbf{F}^{\max}(\mathbf{M}) = \mathbf{u}^{\max}(\mathbf{M})$. Sendo assim, para um mesmo **M**, $\mathbf{F}^{\max}$ resulta em menor pessimismo quando comparado a $\mathbf{u}^{\max}$, com exceção no caso que são iguais, sendo o pessimismo equivalente nesse caso.

Ressalta-se, também, que o conceito de $\mathbf{F}^{\max}$ pode ser estendido para outras abordagens robustas que não utilizam intervalos para lidar com as incertezas, por exemplo, o *Worst Case Scenario Approximation* (WCSA) proposto em (SOARES *et al.*, 2009a). E, nesse caso, $\mathbf{F}^{\max}$ resultará em soluções robustas na imagem viável, o que não é garantido pelo $\mathbf{u}^{\max}$ ao ser estendido para o WCSA.

Ao utilizar $\mathbf{F}^{\max}$ é mais comum que cada solução robusta tenha como imagem um conjunto de pontos (cuja quantidade será no máximo igual ao valor de n_s), ao invés de um único ponto no espaço dos objetivos. Nesse caso, a fronteira não-dominada poderá ser formada pela união de várias $\mathbf{F}^{\max}$ não-dominadas entre si (cada uma associada a um ponto viável **x**) ao invés de uma única fronteira. A Figura 7-6 ilustra essa situação considerando-se três soluções robustas **a**, **b** e **c**. Os pontos no espaço dos objetivos representam o conjunto **M** de cada solução (pontos em cinza para a solução **a**, pretos para a solução **b** e brancos para a solução **c**), o conjunto de fronteiras não-dominadas entre si obtido via $\mathbf{F}^{\max}$ é constituído pelos pontos cinzas e pretos (referem-se às soluções **a** e **b**) e a fronteira não-dominada obtida via $\mathbf{u}^{\max}$ é representada pelos asteriscos pretos. Para que seja possível comparar as diversas $\mathbf{F}^{\max}$ obtidas a fim de determinar quais são não-dominadas entre si, faz-se necessário definir o conceito de dominância entre conjuntos de pontos não-dominados[9].

Definição 7-4 *Seja* $\mathbf{A} \subseteq \mathbf{Z}$ e $\mathbf{B} \subseteq \mathbf{Z}$ *dois conjuntos de pontos não-dominados. Estende-se o operador* $\prec$ *como segue*

$$\mathbf{A} \prec \mathbf{B} \;\Leftrightarrow\; \text{todo } \mathbf{b} \in \mathbf{B} \text{ é dominado por ao menos um } \mathbf{a} \in \mathbf{A}. \tag{7-8}$$

[9] O conceito de conjunto de pontos não dominados é dado pela Definição 4-1.

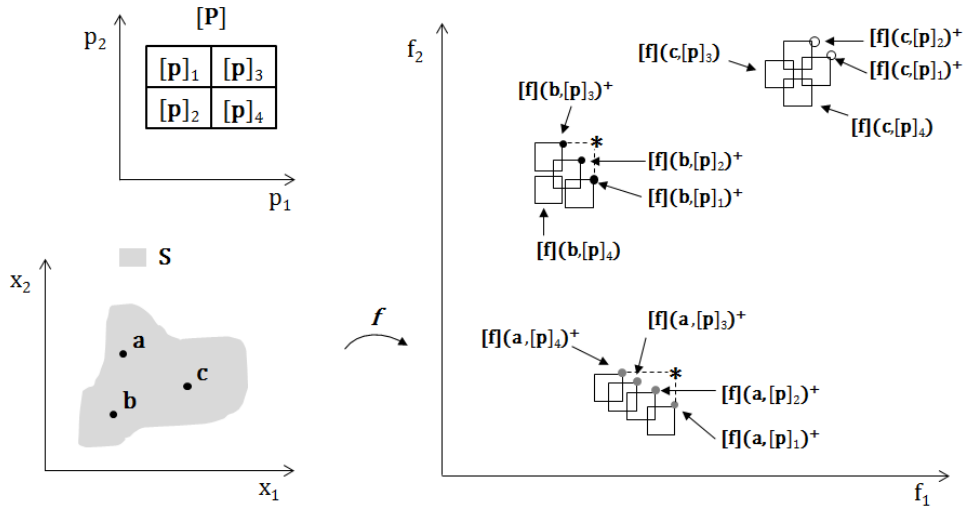


Figura 7-6-Obtenção da fronteira não-dominada para os pontos **a, b** e **c** considerando-se $\mathbf{F}^{\max}$ e $\mathbf{u}^{\max}$ na computação das incertezas em **P**.

A seguir apresentam-se os algoritmos para computação do conjunto **M**, de $\mathbf{u}^{\max}(\mathbf{M})$ e de $\mathbf{F}^{\max}(\mathbf{M})$. O Algoritmo 7-7 computa o conjunto **M** conforme descrito em (7-5). São necessários dois parâmetros de entrada: a solução viável **x** e as caixas intervalares do subpavimento de **P**.

Algoritmo 7-7– Computação do conjunto $\mathbf{M} = [\mathbf{f}](\mathbf{x}, \mathbf{P})^+$.

1. $\mathbf{M} = \emptyset$.
2. Para i = 1 até n_s *//Para cada caixa* $[\mathbf{p}]_i$ *do subpavimento*
   2.1. $\mathbf{M} = \mathbf{M} \cup \{[\mathbf{f}](\mathbf{x}, [\mathbf{p}]_i)^+\}$ *//Inclua* $[\mathbf{f}](\mathbf{x}, [\mathbf{p}]_i)^+$ *ao conjunto* **M**
3. Fim Para
4. Retorne **M**

O Algoritmo 7-8 computa $\mathbf{u}^{\max}(\mathbf{M})$ conforme descrito na Definição 7-2. O conjunto **M** é o único parâmetro de entrada requerido.

Algoritmo 7-8– Computação de $\mathbf{u}^{\max}(\mathbf{M})$.

1. Para k = 1 até n_f *// Para cada objetivo k*
   1.1. $w = M(1{:}n_s)_k$ *// Guarde em w o valor da dimensão k para todos os pontos de* **M**
   1.2. $u_k^{\max} = \max(w)$ *// Guarde o pior valor da dimensão k para todos os pontos de* **M**
2. Fim Para
3. Retorne $\mathbf{u}^{\max}(\mathbf{M})$

O Algoritmo 7-9 é utilizado para computar $\mathbf{F}^{\max}(\mathbf{M})$ de acordo com a Definição 7-3. O único parâmetro de entrada necessário é o conjunto **M**.

Algoritmo 7-9– Computação de $\mathbf{F}^{\max}(\mathbf{M})$.

1. Para j = 1 até n_s *// Para cada ponto j de* **M**
   1.1. Para i = 1 até $|\mathbf{M}|$ *// Para cada ponto i de* **M**
      1.1.1. Se i ≠ j
         1.1.1.1. Se $M(j) \prec M(i)$ *// Se o j-ésimo ponto de* M *domina o i-ésimo ponto de* M
            1.1.1.1.1. Remove M(j) de **M**.
            1.1.1.1.2. Vá para o passo 1.2. *// Avança para o próximo j*
   1.2. Fim Para
2. Fim Para
3. $\mathbf{F}^{\max} = \mathbf{M}$
4. Retorne $\mathbf{F}^{\max}(\mathbf{M})$.

7.3 *Interval Robust Multi-Objective Evolutionary Algorithm – Minimax*

Nesta seção, descreve-se o algoritmo evolucionário proposto para resolver o problema de otimização robusta estabelecido na Subseção 2.1.1. O algoritmo é denominado *Interval Robust Multi-Objective Evolutionary Algorithm – Minimax* (IRMOEA-M). Sua estrutura geral assemelha-se com aquela encontrada nos algoritmos evolucionários multiobjetivo tradicionais, pois apresenta os mecanismos de seleção, cruzamento, mutação e elitismo. No entanto, o IRMOEA-M possui uma característica peculiar que é a incorporação de mecanismos capazes de realizar o tratamento de incertezas pela abordagem robusta *minimax* via análise intervalar.

7.3.1 Estrutura Geral do IRMOEA-M

Inicialmente, apresentam-se as características gerais do algoritmo.

Espaço de incerteza: A formulação do problema a ser resolvido deve descrever como cada incerteza age no problema, além disso, o intervalo de variação de cada um dos parâmetros de incerteza deve ser fornecido. Dessa forma, constrói-se um subpavimento para o espaço de incerteza do problema de acordo com Algoritmo 7-6.

População inicial: Para representação dos indivíduos foi utilizada a codificação real. A população inicial é formada por pontos escolhidos aleatoriamente no domínio permitido para cada uma das variáveis de decisão.

Avaliação dos indivíduos: Realizada via operações intervalares a fim de avaliar o indivíduo frente a cada uma das caixas do subpavimento de incerteza. Portanto, requer a função de inclusão para cada uma das funções objetivo do problema. Na avaliação do indivíduo, o processo de maximização da influência das incertezas pode ser feito por meio do ponto ideal de maximização $\mathbf{u}^{\max}$ ou da fronteira ideal de maximização $\mathbf{F}^{\max}$, o que deve ser definido previamente, pois há peculiaridades no uso de cada um (e.g., a forma de computar dominância e distância da multidão).

Tratamento de restrições: Realizado por meio da avaliação intervalar das funções de restrição, sendo assim, faz-se necessário ter a função de inclusão para cada uma das restrições do problema. O Algoritmo 7-10 descreve o processo de tratamento de restrições, que recebe como entrada um indivíduo **x** e as caixas do subpavimento de incerteza. Como saída tem-se o valor das violações para as restrições, se o valor for zero significa que **x** é factível, pois nenhuma restrição é violada para nenhuma das caixas do subpavimento de incerteza. No Algoritmo 7-10, para um dado **x** e para cada restrição, é verificado se o limite superior do intervalo de saída dessa restrição é menor ou igual a zero, considerando-se todas as caixas do subpavimento. Nesse caso, não ocorre violação, caso contrário, há violação da restrição. Dessa forma, o indivíduo **x** é penalizado de acordo com o nível de violação em cada restrição pela expressão presente no passo 4 do algoritmo. A violação das restrições é o primeiro critério utilizado para fazer o *ranking* dos indivíduos. Quanto maior a violação das restrições, pior é o *ranking* do indivíduo.

Algoritmo 7-10– Tratamento de restrições para o indivíduo **x**.

1. funcao violacao = tratamento_restricoes ($\mathbf{x}$, $\{[\mathbf{p}]_1, ..., [\mathbf{p}]_{n_s}\}$)
2. Para $j = 1$ até n_g *// Para cada restrição j do problema.*
   2.1. Para $i = 1$ até n_s *// Para cada caixa* $[\mathbf{p}]_i$ *do subpavimento de incerteza.*
   2.2. $w_i = [g_j](\mathbf{x}, [\mathbf{p}]_i)^+$ *// Guarda em* w_i *o pior valor da restrição j para cada caixa* $[\mathbf{p}]_i$.
   2.3. Fim Para.
   2.4. $pior\_g_j = \max(\mathbf{w})$; *// Guarda o pior valor da restrição j considerando todo o*
3. Fim Para. *// subpavimento de incerteza dado por* $\{[\mathbf{p}]_1, ..., [\mathbf{p}]_{n_s}\}$.
4. $violacao = 1000 \times \left(\sum_{j=1}^{n_g} \max(0, pior\_g_j)\right)^2$ *// Calcula a violação das restrições.*
5. fim funcao

Ordenação por não dominância: Realizada pela divisão em fronteiras de não dominância como proposto no algoritmo NSGA-II (DEB *et al.*, 2002). Se a maximização da influência das incertezas é feita pelo ponto ideal de maximização $\mathbf{u}^{\max}$, utiliza-se a Definição 2-4 para verificar a dominância. Caso, se use a fronteira ideal de maximização $\mathbf{F}^{\max}$ a dominância é verificada pela Definição 7-4.

Distância da multidão: Feita conforme proposto no algoritmo NSGA-II (DEB *et al.*, 2002). Todavia, se a maximização da influência das incertezas é feita pela fronteira ideal de maximização $\mathbf{F}^{\max}$, há necessidade de reduzir cada fronteira não-dominada pertencente a $\mathbf{F}^{\max}$ para um único ponto de referência antes de aplicar o algoritmo de distância da multidão. Nesta tese, definiu-se tal ponto de referência como sendo o valor médio em cada uma das dimensões considerando-se todos os pontos que compõem a fronteira não-dominada em questão.

Ranking dos indivíduos: Realizado de acordo com a violação das restrições, ordenação por não dominância e distância da multidão. Inicialmente, faz-se o tratamento das restrições. Em seguida, realiza-se a ordenação por não dominância dos indivíduos factíveis. Por último, computa-se a distância da multidão para indivíduos pertencentes a uma mesma fronteira de não dominância.

Elitismo e geração de novos indivíduos: A escolha dos indivíduos que participam das operações evolucionárias foi realizada pelo método de torneio. O *simulated binary crossover* e *polynomial mutation* foram os operadores utilizados no cruzamento e mutação, respectivamente (DEB e AGRAWAL, 1995). A estratégia elitista consistiu em juntar a população corrente com a população filha obtida pela aplicação dos operadores evolucionários e realizar o *ranking* nessa população de acordo com a violação das restrições, dominância e distância da multidão. Feito isso, os indivíduos com melhor *ranking* (até o limite máximo permitido para a população) seguem para a população seguinte.

O IRMOEA-M é descrito no Algoritmo 7-11. Os parâmetros de entrada requeridos pelo IRMOEA-M são: número de gerações, número de indivíduos na população, domínio das variáveis de decisão, tamanho do torneio, processo de maximização da influência das incertezas, intervalo dos parâmetros de incerteza, ε_p que indica o nível de precisão desejado no subpavimento de incerteza, funções de inclusão para as funções objetivo e restrições, probabilidade de cruzamento e probabilidade de mutação.

Algoritmo 7-11– IRMOEA-M.

1. Gere o subpavimento para o espaço de incerteza **P** de acordo com o Algoritmo 7-6.
2. Crie aleatoriamente uma população inicial P_p com n indivíduos.
3. Avalie os n indivíduos de P_p e armazene seus valores de *fitness* em A_p.
4. Faça o *ranking* dos indivíduos.
5. Até que o critério de parada seja satisfeito, repita:
   5.1. Execute o processo de seleção em P_p.
   5.2. Execute os processos de cruzamento e mutação gerando a população filha (P_f).
   5.3. Avalie os indivíduos da população filha e armazene seus valores de *fitness* em A_f.
   5.4. Junte as populações P_p e P_f em P_t e seus respectivos valores de *fitness* A_p e A_f em A_t.
   5.5. Faça o *ranking* dos indivíduos.
   5.6. Mantenha para a população seguinte os n primeiros indivíduos de P_t, armazenando-os em P_p e colocando seus respectivos valores de *fitness* em A_p.
6. Retorne P_p e A_p.

7.4 *Interval Robust Multi-Objective Evolutionary Algorithm – Minimax Regret*

Conforme apresentado no capítulo 3, as abordagens presentes na literatura para lidar com o cenário de pior caso das incertezas em problemas de otimização robusta multiobjetivo utilizam o critério *minimax* (robusto absoluto). As soluções robustas geradas por essas abordagens podem ser entendidas como aquelas que possuem o melhor desempenho frente ao pior cenário de incerteza. Assim, podem ser consideradas como soluções conservadoras em termos de desempenho para outros cenários diferentes do cenário de pior caso. Além disso, como a ocorrência do cenário de pior caso, em geral, está associada a um acontecimento catastrófico, tem-se que a viabilidade das soluções, nesse caso, é mais relevante do que a qualidade dessas. Por exemplo, se em uma indústria houver um problema no sistema de refrigeração e a temperatura do ambiente operacional se elevar consideravelmente. O maquinário presente nesse ambiente pode ter o seu desempenho severamente comprometido, mas será algo momentâneo, até que o problema de refrigeração seja sanado. O importante não é que o maquinário funcione da melhor forma possível nessa situação, mas sim, que não ocorra uma catástrofe, como a explosão ou quebra do maquinário devido à elevação da temperatura. Diante do exposto, é importante obter soluções que tenham bom desempenho nos mais diversos cenários de incerteza.

Uma forma de se obter soluções com bom desempenho em diversos cenários é utilizando-se o *minimax regret* (desvio robusto) discutido na Subseção 3.1.2. Nessa abordagem, o objetivo é encontrar soluções robustas que tenham bom desempenho (menor arrependimento em relação à melhor solução possível para cada um dos cenários estabelecidos) em diferentes cenários de incerteza e que sejam factíveis em todos eles. Inclusive, no cenário de pior caso. Sendo assim, as soluções obtidas pelo *minimax regret* lidam adequadamente com a possibilidade da ocorrência de catástrofes e tendem a possuir menor conservadorismo em relação às obtidas com o robusto absoluto, pois funcionarão para todos os cenários de incerteza com o menor arrependimento possível.

Uma vantagem do *minimax regret* é que o arrependimento máximo funciona como um indicador de quanto a qualidade da solução pode ser melhorada, se as fontes de incerteza do problema fossem

tratadas e solucionadas. Por exemplo, se um problema possui um baixo arrependimento máximo, significa que a incerteza existente no problema não o influencia muito. Sendo assim, talvez não compense investir em formas de eliminar as fontes de incertezas. Por outro lado, em um problema em que o arrependimento máximo for alto, os investimentos para amenizar as incertezas podem levar a soluções bem melhores. Como desvantagem do *minimax regret* tem-se o custo computacional de acordo com a quantidade de cenários de incerteza, uma vez que para se determinar o arrependimento é necessário calcular a melhor solução para cada cenário de incerteza.

O IRMOEA-MR, algoritmo evolucionário proposto para resolver o problema de otimização robusta estabelecido na Subseção 2.1.2, é apresentado no Algoritmo 7-12. Sua estrutura geral assemelha-se ao IRMOEA-M, porém há diferenças que são explicadas a seguir. A primeira delas é a existência de duas etapas adicionais (passos 2 e 3), que envolvem a obtenção da melhor solução para cada caixa pertencente ao subpavimento do espaço de incerteza **P**, bem como a redução dessa melhor solução a um único ponto de referência, respectivamente. A etapa 2 pode ser realizada de duas formas distintas: i) uso de um algoritmo robusto multiobjetivo ou ii) uso de um algoritmo robusto mono-objetivo. Nesta tese, a primeira forma é feita executando-se uma vez o IRMOEA-M, para cada caixa do subpavimento do espaço de incerteza **P**, o que resultaria em n_s execuções do IRMOEA-M. Após cada execução, para realizar a etapa 3 basta, conforme definido previamente pela Equação (2-12), computar o ponto de referência da melhor solução obtida. Como resultado, n_s pontos de referência são obtidos ao fim do processo. Para realizar pela segunda forma, nesta tese, propõe-se o *Interval Robust Evolutionary Algorithm – Minimax* (IREA-M), que é um algoritmo robusto mono-objetivo descrito pelo Algoritmo 7-13. Nessa segunda forma, o IREA-M é executado n_f vezes para cada uma das caixas do subpavimento de incerteza. A cada execução, para uma dada caixa do subpavimento, resolve-se o problema de otimização para um dos objetivos em particular. Após resolver para todos os n_f objetivos em separado, é possível calcular o ponto de referência, também pela Equação (2-12), considerando-se todos os objetivos simultaneamente. Como há n_s caixas no subpavimento de incerteza, são necessárias $n_f \times n_s$ execuções do IREA-M para a obtenção dos n_s pontos de referência.

Algoritmo 7-12– IRMOEA-MR.

1. Gere o subpavimento para o espaço de incerteza **P** de acordo com o Algoritmo 7-6.
2. Encontre a melhor solução para cada caixa do subpavimento de **P**.
3. Encontre o ponto de referência da melhor solução de cada caixa do subpavimento de **P** conforme a Equação (2-12).
4. Crie aleatoriamente uma população inicial P_p com n indivíduos.
5. Avalie os n indivíduos de P_p e armazene seus valores de *fitness* (computado via operador de *regret* definido pela Equação (2-13) e redefinido pela Equação (7-9) para o tratamento intervalar das incertezas) em A_p.
6. Faça o *ranking* dos indivíduos.
7. Até que o critério de parada seja satisfeito, repita:
   7.1. Execute o processo de seleção em P_p.
   7.2. Execute os processos de cruzamento e mutação gerando a população filha (P_f).
   7.3. Avalie os indivíduos da população filha e armazene seus valores de *fitness* (computado via operador de *regret*) em A_f.
   7.4. Junte as populações P_p e P_f em P_t e seus respectivos valores de *fitness* A_p e A_f em A_t.
   7.5. Faça o *ranking* dos indivíduos.
   7.6. Mantenha para a população seguinte os n primeiros indivíduos de P_t, armazenando-os em P_p e colocando seus respectivos valores de *fitness* em A_p.
8. Retorne P_p e A_p.

A segunda diferença é relativa à avaliação dos indivíduos, uma vez que o IRMOEA-MR utiliza o operador de *regret* $\ominus$ definido em (2-13), o qual é redefinido em sequência para $\boldsymbol{f}(\mathbf{x}, [\mathbf{P}]) \ominus \boldsymbol{f}^*([\mathbf{P}])$ dado o tratamento intervalar das incertezas:

$$\mathbf{r} = \begin{pmatrix} max\left([\mathrm{f}]_1(\mathbf{x}, [\mathbf{p}]_1)^+ - \mathrm{v}_1(\mathbf{V}_1), \dots, [\mathrm{f}]_1(\mathbf{x}, [\mathbf{p}]_\mathrm{j})^+ - \mathrm{v}_1(\mathbf{V}_\mathrm{j}), \dots, [\mathrm{f}]_1(\mathbf{x}, [\mathbf{p}]_{n_i})^+ - \mathrm{v}_1(\mathbf{V}_{n_i})\right), \dots, \\ max\left([\mathrm{f}]_k(\mathbf{x}, [\mathbf{p}]_1)^+ - \mathrm{v}_\mathrm{k}(\mathbf{V}_1), \dots, [\mathrm{f}]_k(\mathbf{x}, [\mathbf{p}]_\mathrm{j})^+ - \mathrm{v}_\mathrm{k}(\mathbf{V}_\mathrm{j}), \dots, [\mathrm{f}]_k(\mathbf{x}, [\mathbf{p}]_{n_i})^+ - \mathrm{v}_\mathrm{k}(\mathbf{V}_{n_i})\right), \dots, \\ max\left([\mathrm{f}]_{n_f}(\mathbf{x}, [\mathbf{p}]_1)^+ - \mathrm{v}_{n_f}(\mathbf{V}_1), \dots, [\mathrm{f}]_{n_f}(\mathbf{x}, [\mathbf{p}]_\mathrm{j})^+ - \mathrm{v}_{n_f}(\mathbf{V}_\mathrm{j}), \dots, [\mathrm{f}]_{n_f}(\mathbf{x}, [\mathbf{p}]_{n_i})^+ - \mathrm{v}_{n_f}(\mathbf{V}_{n_i})\right) \end{pmatrix}. \quad (7\text{-}9)$$

A terceira diferença ocorre no procedimento de ordenação por não dominância, pois no IRMOEA-MR a dominância é verificada exclusivamente pela Definição 2-4.

Sendo assim, os itens relacionados ao espaço de incerteza, população inicial, tratamento de restrições, distância da multidão (exceto a parte sobre a fronteira ideal de maximização), elitismo e geração de novos indivíduos descritos na Subseção 7.3.1 também são válidos para o IRMOEA-MR.

Algoritmo 7-13– IREA-M.

1. Crie aleatoriamente uma população inicial P_p com n indivíduos.
2. Pegue uma única caixa [**p**] do subpavimento para o espaço de incerteza **P**.
3. Avalie cada indivíduo **x** de P_p por meio de $[\mathrm{f}](\mathbf{x}, [\mathbf{p}])$ e armazene $[\mathrm{f}](\mathbf{x}, [\mathbf{p}])^+$ como seu valor de *fitness*.
4. Penalize o valor de *fitness* de cada indivíduo de acordo com a sua violação das restrições conforme o Algoritmo 7-10 aplicado somente à caixa [**p**].
5. Até que o critério de parada seja satisfeito, repita:
   5.1. Execute o processo de seleção (torneio) em P_p.
   5.2. Execute os processos de cruzamento e mutação gerando a população filha (P_f).
   5.3. Avalie os indivíduos da P_f e armazene seus valores de *fitness* (da mesma forma que no passo 3).
   5.4. Penalize o valor de *fitness* de cada indivíduo da P_f de acordo com a sua violação das restrições conforme o Algoritmo 7-10 aplicado somente à caixa [**p**].
   5.5. Junte as populações P_p e P_f em P_t e seus respectivos valores de *fitness* A_p e A_f em A_t.
   5.6. Selecione para a população seguinte n indivíduos de P_t (sendo um percentual desses constituído pelos melhores), armazenando-os em P_p e colocando seus respectivos valores de *fitness* em A_p.
6. Retorne o indivíduo com o melhor valor de *fitness*.

A Figura 7-7 ilustra as etapas 2 e 3 do IRMOEA-MR. Assume-se que o espaço de incertezas é bisseccionado em quatro caixas e que o IRMOEA-M é utilizado na etapa 2 do Algoritmo 7-12 para resolver o problema descrito em (2-11), resultando nos conjuntos $\mathbf{V_1}, \mathbf{V_2}, \mathbf{V_3}$ e $\mathbf{V_4}$. A definição dada pela Equação (2-12) é utilizada para determinar os pontos de referência $\mathbf{v}(\mathbf{V_1}), \mathbf{v}(\mathbf{V_2}), \mathbf{v}(\mathbf{V_3})$ e $\mathbf{v}(\mathbf{V_4})$ de cada caixa do subpavimento de incerteza, os quais são representados por asteriscos pretos (exceto $\mathbf{v}(\mathbf{V_4})$, pois esse é idêntico a $\mathbf{V_4}$ que só possui um elemento). Para ilustrar o uso do operador de *regret* $\ominus$, assume-se a existência de uma solução $\mathbf{x}$. Assim, de acordo, com (7-9) tem-se que $\mathbf{r}$ é:

$$\begin{pmatrix} max\left([\mathrm{f}]_1(\mathbf{x},[\mathbf{p}]_1)^+ - \mathrm{v}_1(\mathbf{V}_1), [\mathrm{f}]_1(\mathbf{x},[\mathbf{p}]_2)^+ - \mathrm{v}_1(\mathbf{V_2}), [\mathrm{f}]_1(\mathbf{x},[\mathbf{p}]_3)^+ - \mathrm{v}_1(\mathbf{V_3}), [\mathrm{f}]_1(\mathbf{x},[\mathbf{p}]_4)^+ - \mathrm{v}_1(\mathbf{V_4})\right), \\ max\left([\mathrm{f}]_2(\mathbf{x},[\mathbf{p}]_1)^+ - \mathrm{v}_2(\mathbf{V}_1),\ [\mathrm{f}]_2(\mathbf{x},[\mathbf{p}]_2)^+ - \mathrm{v}_2(\mathbf{V_2}), [\mathrm{f}]_2(\mathbf{x},[\mathbf{p}]_3)^+ - \mathrm{v}_2(\mathbf{V_3}), [\mathrm{f}]_2(\mathbf{x},[\mathbf{p}]_4)^+ - \mathrm{v}_2(\mathbf{V_4})\right) \end{pmatrix}. \quad (7\text{-}10)$$

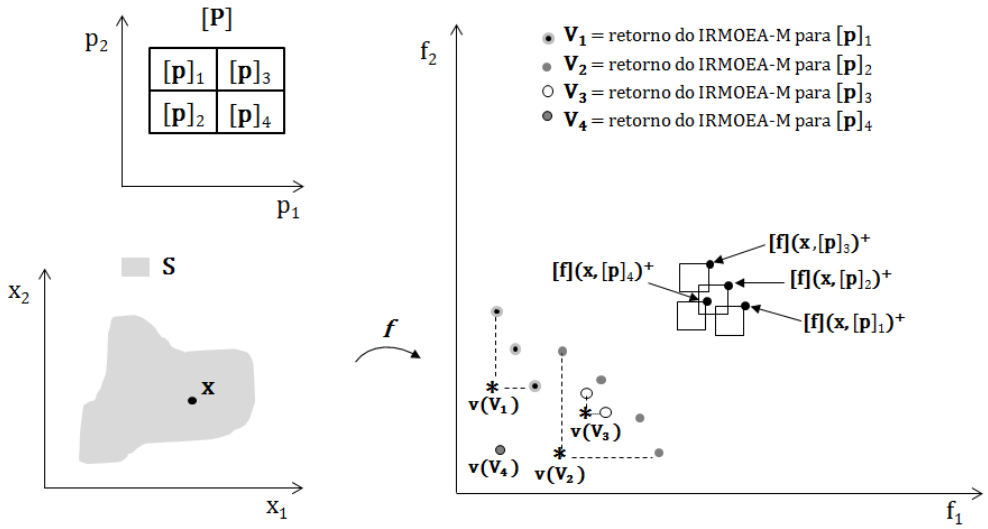


Figura 7-7-Obtenção dos pontos de referência para o subpavimento de **P** no IRMOEA-MR.

7.5 Conclusão

Neste capítulo foram propostos dois algoritmos baseados em GP com a finalidade de se determinar expressões analíticas que gerem boas funções de inclusão a partir da aplicação do Teorema 2. Em seguida, descreveu-se o ponto ideal de maximização, que é uma forma de maximizar a ação das incertezas em problemas de otimização utilizando-se caixas intervalares. Em sequência, a fronteira ideal de maximização foi proposta e definida. Além disso, foram apresentados os algoritmos para a computação de $\mathbf{u}^{\max}$ e $\mathbf{F}^{\max}$. Também, foi proposto o algoritmo evolucionário IRMOEA-M, o qual incorpora mecanismos capazes de realizar o tratamento de incertezas pela abordagem robusta *minimax* via análise intervalar. Por fim, a noção de robustez *minimax regret* foi discutida como uma alternativa interessante para encontrar as soluções que funcionem no pior caso e que nos outros cenários não sejam conservadoras. A noção de robustez *minimax regret* foi implementada no algoritmo proposto IRMOEA-MR. Os algoritmos, definições e formulações propostas neste capítulo buscaram apresentar soluções para as principais dificuldades observadas na literatura no que diz respeito à otimização robusta com IA.

8. EXPERIMENTOS E RESULTADOS

8.1 Aproximação de Funções de Inclusão em Problemas de Engenharia de Controle

De acordo com ASTROM e HAGGLUND (2001), o controlador Proporcional-Integral-Derivativo (PID) vem sendo amplamente utilizado nos processos industriais devido a sua estrutura simples e o bom desempenho em diversas condições operacionais. ASTROM e HAGGLUND (2004) indicam que uma pesquisa feita em mais de 11000 controladores regulatórios, 97% desses possuía a estrutura PID. Em poucas palavras, um controlador PID tenta minimizar o erro "e" entre uma variável medida y(t) e um *setpoint* desejado "r" por meio do ajuste de três parâmetros: o ganho proporcional k_p, o ganho integral k_i e o ganho derivativo k_d (LI *et al.*, 2006). De acordo com GAING (2004), a função de transferência de um controlador PID no domínio da frequência é dada por:

$$C(s) = k_p + \frac{k_i}{s} + k_d s\,. \tag{8-1}$$

O processo de encontrar a melhor configuração dos parâmetros de ganho é chamado de *tuning* ou sintonia. O desempenho de um controlador mal ajustado é limitado, sendo assim, é interessante dispor de métodos para ajustar o controlador. Pode-se até mesmo usar procedimentos de tentativa e erro, no entanto, o tempo computacional requerido é alto. Diante disso, observa-se na literatura o uso de métodos mais eficientes que consistem em formular o PID *tuning* como um problema de otimização, o qual tem como objetivo uma ou mais características da resposta ao degrau unitário e medidas de desempenho relacionadas ao erro no estado estacionário. As características de resposta ao degrau unitário podem ser visualizadas na Figura 8-1. São elas: tempo de subida, sobre-sinal máximo, tempo de pico e tempo de acomodação.

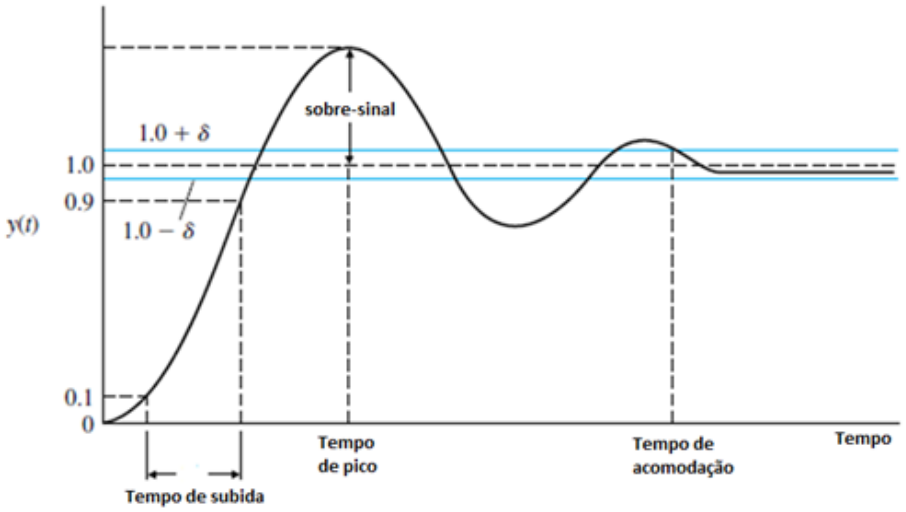


Figura 8-1 - Características da Resposta ao Degrau Unitário. Adaptada de (OGATA, 1997).

De acordo com OGATA (1997) têm-se as seguintes definições para as características de resposta ao degrau: 1) O tempo de subida é o tempo requerido para que a resposta suba de 10% até 90% (usado neste trabalho), 5% a 95% ou 0% a 100% em relação ao valor de referência. 2) O tempo de pico é o tempo gasto para que a resposta alcance o primeiro pico do sobre-sinal. 3) O sobre-sinal máximo é a diferença entre o valor mais alto da resposta e o valor de referência. 4) O tempo de acomodação é o tempo necessário para que a curva de resposta fique dentro de um faixa de variação admissível em relação ao valor de referência de, em geral, 2% (usado neste trabalho) ou 5% de variação.

As medidas de desempenho estão relacionadas às seguintes métricas de erro no estado estacionário: integral do erro absoluto (IAE), integral do erro quadrático (ISE) e integral do erro absoluto ponderado pelo tempo (ITAE), os quais são definidos em (8-2), (8-3) e (8-4).

$$\mathrm{IAE} = \int_0^\infty |\mathrm{e(t)}|\mathrm{dt} \tag{8-2}$$

$$\mathrm{ISE} = \int_0^\infty \mathrm{e^2(t)dt} \tag{8-3}$$

$$\mathrm{ITAE} = \int_0^\infty \mathrm{t}|\mathrm{e(t)}|\mathrm{dt} \tag{8-4}$$

A seguir, são apresentados alguns trabalhos em que o PID *tuning* é formulado como um problema de otimização. GAING (2004) emprega *swarm optimization* para buscar a configuração ótima de parâmetros PID de um sistema de regulador de tensão automático. TAKAHASHI *et al*. (1997) apresentam uma metodologia para projeto de compensadores PID para sistemas sujeitos a sinais de perturbação e incertezas paramétricas. PEDERSEN *et al.* (2006) investigam o PID *tuning* para um sistema de levitação magnética usando o algoritmo NSGA-II e JINHUA *et al.* (2009) propõem um algoritmo genético auto-organizado que é usado para ajustar os parâmetros de ganho do controlador PID. Em particular, SOARES (2008) propõe uma formulação multiobjetivo robusta que usa análise intervalar para lidar com incertezas nos parâmetros de ganho k_p e k_d. Na abordagem por IA, é essencial obter funções de inclusão em termos dos parâmetros de ganho. Essas funções de inclusão são obtidas a partir das expressões analíticas para cada uma das características de resposta ao degrau. A dificuldade de obtenção de tais expressões analíticas cresce de acordo com a complexidade do modelo matemático da função de transferência em malha fechada. Para mitigar esta dificuldade, foram aplicados os algoritmos SNIF-GPA e SNIF-MOGPA para resolver regressões simbólicas e obter as funções de inclusão aproximadas. Nas subseções 8.1.1 e 8.1.2 o objetivo é descobrir funções de inclusões para as características de resposta ao degrau para o problema de posicionamento do motor de corrente contínua e, em sequência, na Subseção 8.1.3 é realizada a sintonia robusta do controlador. Na Subseção 8.1.4 aplica-se o SNIF-GPA em um problema de controle adaptado de (SOARES, 2008) e, por fim, na Subseção 8.1.5 realiza-se a sintonia robusta do controlador relativo a esse problema.

8.1.1 Aplicação do SNIF-GPA ao Problema do Motor de Corrente Contínua

Esta subseção baseia-se no trabalho desenvolvido em (MENDES *et al.*, 2010). O motor de corrente contínua é um atuador comum em sistemas de controle. Ele fornece diretamente movimento rotativo e, juntamente com rodas ou tambores e cabos, pode fornecer movimento de translação. O objetivo principal é ajustar a posição do motor precisamente por meio de um controlador PID. As especificações do problema foram baseadas em TILBURY *et al.* (1998) e os valores dos parâmetros são apresentados na Tabela 8-1.

Tabela 8-1 - Valores dos parâmetros físicos para o problema de posição do motor de corrente contínua.

Parâmetro físico (abreviatura)	Valor
Momento de inércia do rotor (J)	3,2284E-6 kg.m2
Razão de amortecimento do sistema mecânico (B)	3,5077E-6 Nms
Constante de força eletromotriz (K)	0,0274 Nm/Amp
Resistência Elétrica (R)	4 Ω
Indutância Elétrica (L)	2,75E-6 H

A função de transferência do sistema é dada pela Equação (8-5):

$$\frac{\theta(s)}{V(s)} = \frac{K}{s\left((Js + B)(Ls + R) + K^2\right)} \tag{8-5}$$

sendo a posição do eixo (θ) a saída do sistema e a tensão (V) a entrada. A Equação (8-5) é de terceira ordem.

Os parâmetros utilizados pelo algoritmo são apresentados na Tabela 8-2. A escolha dos elementos do conjunto de funções foi feita de modo a gerar expressões que contenham somente operadores elementares e funções contínuas, o que é desejável em funções de inclusão.

Tabela 8-2 – Parâmetros do SNIF-GPA.

Parâmetro	Valor
Número máximo de gerações	50
Tamanho da população	500, 1000 ou 2000
Taxa de cruzamento, mutação e reprodução	0,9, 0,1 e 0,2, respectivamente
Taxa de cruzamento entre nós internos	0,8
Conjunto de funções	+, -, *, /, ^, √, seno, ln e exp
Conjunto de terminais	k_p, k_i, k_d e constante efêmera
Intervalo das constantes efêmeras reais e inteiras	]-10, 10[e {1, 2, …, 5}
Tamanho do torneio	2
Altura máxima da árvore durante a execução	7
Altura máxima da árvore na geração inicial	2
Método de geração da população inicial	*Ramped half-and-half*
Critério de parada	Número máximo de gerações
Medida de *fitness*	MSE com *fitness scaling*
Conjunto de treinamento - *fitness cases*	900 amostras
Conjunto de validação	900 amostras divididas em 9 grupos

As amostras foram obtidas utilizando-se rotinas numéricas próprias do Matlab®, como mostrado por TILBURY *et al.* (1998). Executaram-se 1800 simulações com valores aleatórios dentro do mesmo intervalo $]0, 1000[$ para todos os três parâmetros de ganho. Os dados foram igualmente divididos entre os conjuntos de treinamento e validação. Além disso, o conjunto de validação foi subdividido em nove grupos, cada um com cem amostras. O SNIF-GPA foi executado vinte vezes para cada característica da resposta ao degrau unitário. Utilizou-se tamanho da população igual a mil para o tempo de subida, dois mil para o tempo de acomodação e ITAE, e quinhentos para o sobre-sinal e o tempo de pico.

A população da última geração de cada execução foi submetida ao algoritmo de validação, considerando cada um dos nove grupos. Com isso, obteve-se para cada expressão a média da tolerância (no contexto do SNIF-GPA e do SNIF-MOGPA, o conceito de tolerância foi definido na Subseção 7.1.2.3) e a média do MSE (definição do MSE foi dada na Seção 6.3 pela Equação (6.5)). Finalmente, a expressão com a menor média da tolerância foi considerada a melhor. A Tabela 8-3, a Tabela 8-4 e a Tabela 8-5 fornecem os resultados para a melhor expressão encontrada para cada característica de resposta ao degrau e para o ITAE. O algoritmo foi executado em um computador com CPU i3 e 4 GB de memória RAM.

Tabela 8-3 – MSE da melhor expressão obtida para cada característica na fase de treinamento.

Características da resposta ao degrau	MSE na fase de Treinamento
Tempo de subida	4,96E-10
Sobre-sinal	4,06E-5
Tempo de acomodação	6,66E-12
Tempo de pico	1,38E-9
ITAE	1,09E-6

De acordo com a Tabela 8-3, pode-se verificar que a expressão para o tempo de acomodação obteve o menor valor de *fitness* no treinamento quando comparada às expressões para as outras características. A Tabela 8-4 apresenta o tempo de execução requerido pelo SNIF-GPA para a obtenção da melhor expressão para cada característica da resposta ao degrau.

Tabela 8-4 – Tempo de execução necessário pelo SNIF-GPA para a obtenção da melhor expressão em cada característica.

Características da resposta ao degrau	Tempo de execução (s)
Tempo de subida	510
Sobre-sinal	376
Tempo de acomodação	1740
Tempo de pico	240
ITAE	1371

A Tabela 8-5 mostra o MSE da melhor expressão de cada característica para cada um dos nove conjuntos de validação. Verifica-se que em média a expressão obtida para o tempo de subida tem o menor erro e que a expressão obtida para o sobre-sinal possui o maior valor de erro.

Tabela 8-5 – MSE da melhor expressão obtida para cada característica na fase de validação.

Característica da resposta ao degrau	MSE da melhor expressão obtida calculado sobre os pontos de validação										
	Conjunto de validação									Média	Mediana
	1	2	3	4	5	6	7	8	9		
Tempo de subida	2,42E-11	1,13E-12	4,38E-09	2,60E-09	1,50E-10	8,64E-11	1,51E-10	1,90E-08	9,76E-11	2,94E-9	1,50E-10
Sobre-sinal	2,65E-05	3,33E-05	0,000257	0,003463	3,68E-05	3,85E-05	3,18E-05	0,000657	2,81E-05	5,08E-4	3,68E-5
Tempo de acomodação	7,27E-12	3,42E-12	1,57E-07	2,98E-06	6,12E-11	7,82E-13	8,23E-12	8,53E-08	2,12E-12	3,57E-7	8,23E-12
Tempo de pico	1,46E-09	3,33E-09	6,16E-10	2,80E-08	7,71E-08	4,52E-09	1,59E-09	2,56E-09	3,45E-08	1,71E-8	3,33E-9
ITAE	1,72E-06	1,10E-07	6,91E-09	3,69E-06	0,002119	1,67E-06	5,98E-07	1,41E-06	1,75E-05	2,38E-4	1,67E-6

Na Tabela 8-6 são apresentados os valores de tolerância para cada um dos nove conjuntos de validação, considerando-se cada uma das características de resposta ao degrau. Nota-se que os valores da média da tolerância estão compreendidos entre 1,45% e 6,22%, sendo o menor valor para o sobre-sinal e o maior valor para o tempo de pico.

Tabela 8-6 – Tolerância da melhor expressão obtida para cada característica na fase de validação.

Característica da resposta ao degrau	Tolerância da melhor expressão obtida (%)										
	Conjunto de validação									Média	Desvio-padrão
	1	2	3	4	5	6	7	8	9		
Tempo de subida	3,83	4,47	4,47	4,92	4,13	5,24	4,42	4,67	4,18	4,48	0,42
Sobre-sinal	1,13	0,98	3,40	0,97	0,97	3,11	0,88	0,65	0,96	1,45	1,03
Tempo de acomodação	8,87	5,85	8,11	4,45	6,39	3,85	5,87	4,24	5,40	5,89	1,70
Tempo de pico	5,87	5,69	5,52	7,02	5,13	8,06	6,10	6,52	6,10	6,22	0,88
ITAE	4,42	4,46	4,70	8,37	4,58	3,47	8,70	4,18	3,41	5,14	1,98

As expressões (8-6), (8-7), (8-8), (8-9) e (8-10) referem-se às melhores expressões (com algumas simplificações matemáticas) obtidas para o tempo de subida (t_s), sobre-sinal (SS), tempo de acomodação (t_a), tempo de pico (t_p) e ITAE, respectivamente. De acordo com a teoria de sistemas de controle existem efeitos individuais de cada um dos parâmetros de ganho sobre as características de resposta ao degrau. Apesar disso, nota-se que em todas as expressões encontradas há ausência de pelo menos um dos parâmetros. Isso sugere que para os algoritmos de treinamento e validação propostos os efeitos dos parâmetros ausentes não se mostraram significantes quando comparando aos outros tendo em vista as peculiaridades do problema.

$$t_s = \frac{\left(\frac{1}{e^{0,77669}}\right)^{\left(\left(\frac{1}{e^{0,75797}}\right)^{\frac{1}{kd}}\right)^{\frac{1}{kd}} + \frac{\ln(2,23)}{e^{kd}}} + \frac{1}{e^{0,77669}}}{(\ln(2,23)^{0,5789992464})^{\left(\frac{1}{e^{0,77669}}\right)^{\ln(1,678)(e^{0,69874})^{\frac{5}{kd}}}}} \tag{8-6}$$

$$SS = -\frac{58527 \sin\left(\frac{5000\sqrt{kd}}{32829}\right) \ln(kd)^2}{10000} + \frac{17513 \ln(ki+5)}{2000\, kd} + \left(\sqrt{\sqrt{kd}+2}\right)\ln(kd) + KD \tag{8-7}$$

$$t_a = -\frac{4}{\frac{23559\sin(4)\ln(kd)}{10000} + \sqrt{kd} + e^3 + \frac{1}{kd^{2,3559}}} + \frac{\ln(kd) + \sin\left(\frac{94079\,e^3}{30000\,kd}\right) + e^3}{kd} \quad (8\text{-}8)$$

$$t_p = \frac{e^{\frac{\sqrt{2887}}{50}}(\ln(kp) + 2\,kd + 108{,}6219)}{(kd + 1{,}4475)\left(\frac{400\,kd}{979} + 19{,}3877\right)(\ln(kp) + 2\,kd + 101{,}6191)} \quad (8\text{-}9)$$

$$ITAE = \left(\frac{\left((2{,}885) - \left(\sin\left(\frac{7{,}634}{kd}\right)\right)\right)(\sqrt{kp})}{(\sqrt{kd})^{\ln(kd)}}\right) + \left(\frac{9{,}2088}{2{,}7183\,kd}\right) \quad (8\text{-}10)$$

A Tabela 8-7 exibe os valores da inclinação e do intercepto para cada uma das expressões obtidas. Esses valores são calculados durante a fase de treinamento.

Tabela 8-7 – Inclinação e intercepto para as melhores expressões obtidas.

Característica	Inclinação	Intercepto
Tempo de subida	0,009244	-0,009262
Sobre-sinal	0,000187	-0,009856
Tempo de acomodação	0,000082	0,000011
Tempo de pico	0,048152	0,000002
ITAE	0,000499	0,000016

Uma vez obtidas as expressões aproximadas para as características de resposta ao degrau é possível realizar a sintonia robusta do controlador PID associado ao problema do motor de corrente contínua aplicando-se algoritmos de otimização. Um exemplo é mostrado na Subseção 8.1.3.

8.1.2 Aplicação do SNIF-MOGPA ao Problema do Motor de Corrente Contínua

Nesta subseção, apresenta-se o resultado da aplicação do SNIF-MOGPA ao problema de modelagem de posicionamento do motor de corrente contínua, a partir da formulação (7-2). Objetivou-se comparar o efeito entre medidas distintas de complexidade nesse problema. Sendo assim, estabeleceram-se dois esquemas distintos e, em ambos, considerou-se um objetivo para qualidade e outro para complexidade das expressões. O Esquema 1 consiste em utilizar como critérios de otimização o MSE com *fitness scaling* (qualidade) e a altura da árvore (complexidade). O Esquema 2 também usa o MSE com *fitness scaling* para a qualidade das expressões, no entanto, o critério de complexidade é o somatório da quantidade de nós de todas as subárvores.

Assim como feito na subseção anterior, as amostras foram obtidas utilizando-se rotinas numéricas próprias do Matlab®, como mostrado em (TILBURY *et al.*, 1998). Executaram-se 1300 simulações com valores aleatórios no mesmo intervalo$]0, 1000[$ para todos os três parâmetros de ganho. As amostras foram divididas entre o conjunto de treinamento (900 amostras ou 69,23%) e conjunto de validação (400 amostras ou 30,77%). O conjunto de validação foi subdividido em 4 grupos, cada um

com cem amostras. O algoritmo foi executado dez vezes para cada característica de resposta ao degrau em cada esquema. Os parâmetros do SNIF-MOGPA são mostrados na Tabela 8-8.

No que tange a validação, utilizou-se o Algoritmo 7-5, o qual retorna todas as expressões não-dominadas em uma dada execução, considerando-se como critério de *ranking* a média da tolerância na validação e a medida de complexidade da expressão (que depende do esquema). Como o algoritmo é executado dez vezes para cada característica de resposta ao degrau, é necessário unir todas as expressões obtidas em cada execução, realizar novamente o *ranking* e selecionar somente as não-dominadas. Com isso, obtêm-se as expressões que detêm melhor *trade-off* entre a média da tolerância e complexidade, considerando-se todas as dez execuções. A Figura 8-2 mostra os resultados para cada característica de resposta ao degrau de acordo com o esquema.

Tabela 8-8 – Parâmetros do SNIF-MOGPA.

Parâmetro	**Valor**
Número máximo de gerações	200
Tamanho da população	600
Taxa de cruzamento, mutação e reprodução	0,9, 0,1 e 0,2, respectivamente
Taxa de cruzamento entre nós internos	0,8
Conjunto de funções	+, -, *, /, ^, √, seno e exp
Conjunto de terminais	k_p, k_i, k_d e constante efêmera
Intervalo das constantes efêmeras reais e inteiras	]-10, 10[e {1, 2, …, 5}
Tamanho do torneio	6
Altura máxima da árvore durante a execução	10
Altura máxima da árvore na geração inicial	4
Método de geração da população inicial	*Ramped half-and-half*
Critério de parada	Número máximo de gerações
Medida de *fitness*	Esquema 1: MSE com *fitness scaling* e altura da árvore Esquema 2: MSE com *fitness scaling* e somatório da quantidade de nós de todas as subárvores
Conjunto de treinamento - *fitness cases*	900 amostras
Conjunto de validação	400 amostras divididas em 4 grupos

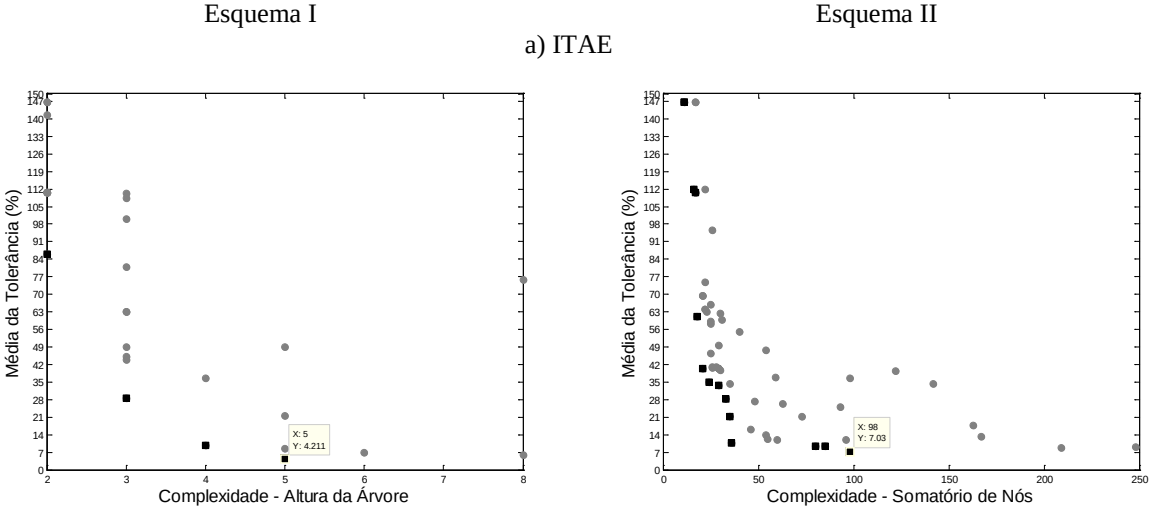
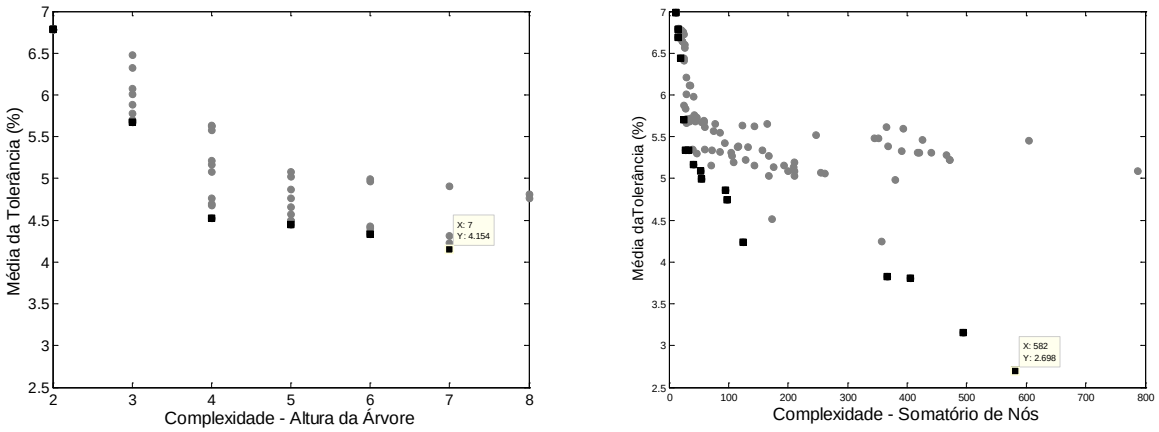


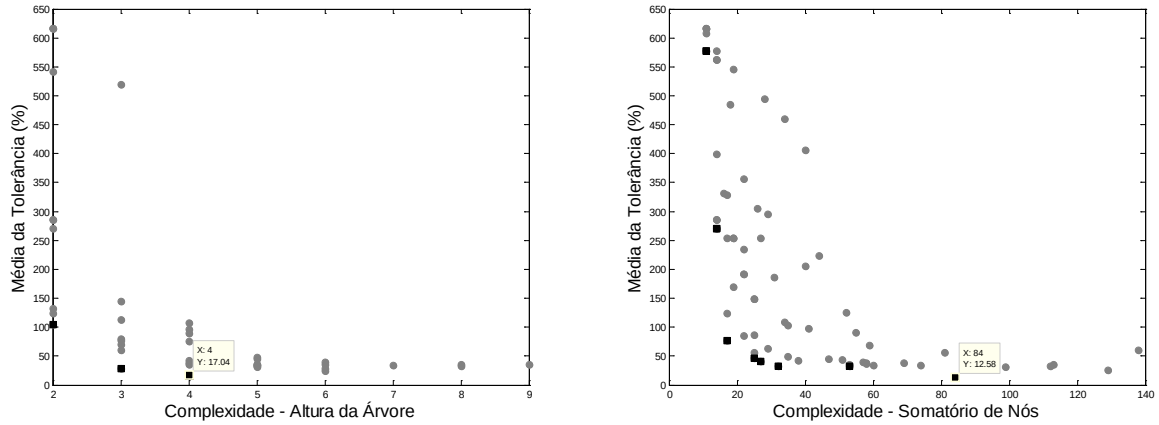
Figura 8-2 – Soluções dominadas e não-dominadas para dez execuções do algoritmo em ambos os esquemas, considerando cada uma das características de resposta ao degrau como segue: a) ITAE, b) Sobre-sinal, c) Tempo de pico, d) Tempo de subida e e) Tempo de acomodação.

Esquema I Esquema II

b) Sobre-sinal



c) Tempo de pico



d) Tempo de subida

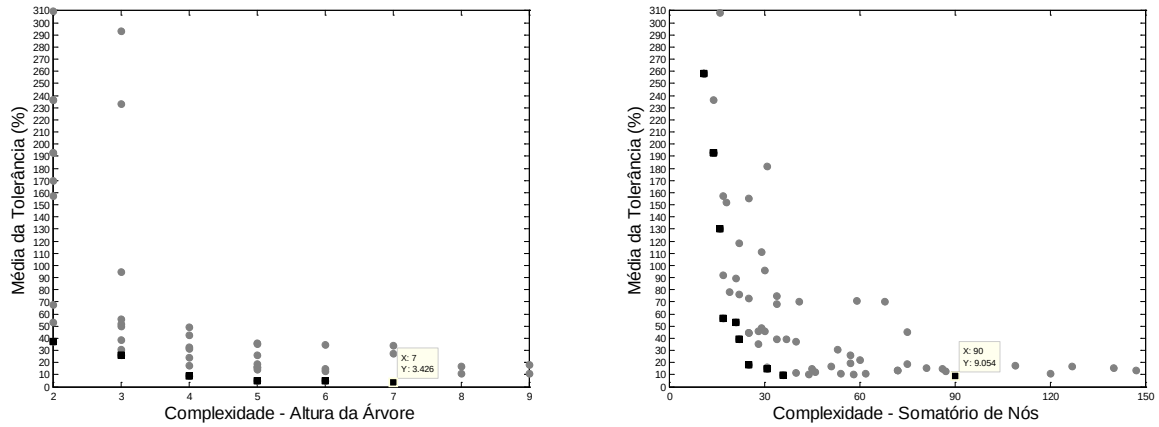


Figura 8-2 – (continuação).

e) Tempo de acomodação

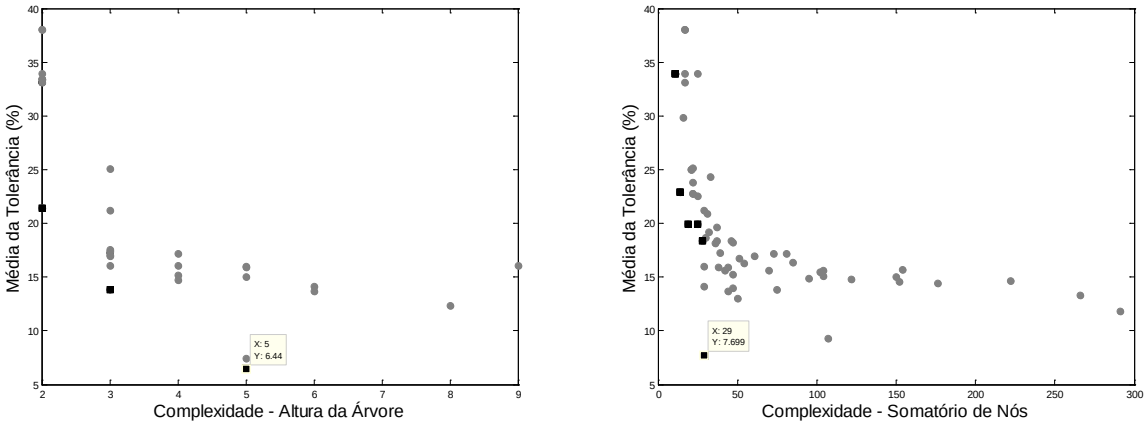


Figura 8-2 – (continuação).

De acordo com a Figura 8-2, nota-se que para cada uma das características de resposta ao degrau a fronteira não-dominada obtida no Esquema II possui mais soluções quando comparada à respectiva fronteira para o Esquema I. Isto é explicado pelo fato do valor da medida de complexidade no Esquema I pertencer ao intervalo de {2, 3, ..., 10}, sendo 10 a maior altura admitida para a árvore. Sendo assim, no Esquema I encontram-se no máximo nove soluções não-dominadas. Percebe-se também que, para todas as características, foram encontradas soluções com a menor altura possível para a árvore (dois). Entretanto, somente para o sobre-sinal é que a média da tolerância (entre 6,5% e 7%) está próxima da melhor obtida (entre 4% e 4,5%). A expressão para o tempo de pico foi a única em que a melhor média da tolerância em ambos os esquemas ficou acima de 10%. Também é possível verificar que para todas as características, excetuando-se o tempo de pico, o melhor valor para a média da tolerância considerando-se os dois esquemas é menor que 7%. Além disso, ao comparar as soluções obtidas no Esquema I e II em termos do menor valor para a média da tolerância, nota-se que os resultados para o Esquema I superam os resultados do Esquema II nas características tempo de subida, tempo de acomodação e ITAE.

A seguir, exibem-se em (8-11) e (8-12) as expressões com melhor média da tolerância obtidas para o ITAE e tempo de subida (t_s), respectivamente. Para melhor apresentação, as expressões foram simplificadas utilizando-se a função *simplify* do Matlab®.

$$\mathrm{ITAE} = \frac{\mathrm{kp}}{\mathrm{kd}^2} + \frac{143907}{10000\mathrm{kd}} + \frac{\mathrm{kp}(\mathrm{kp} - \frac{\mathrm{kp}}{\mathrm{kd}})}{\mathrm{kd}^2(\mathrm{kd} + \mathrm{kp} - \frac{\mathrm{kp}}{\mathrm{ki}})} \tag{8-11}$$

$$t_s = \frac{9\mathrm{kd} + 2\sin(\mathrm{kp}) + \sqrt{\frac{47179}{100}} + \sqrt{\mathrm{ki}} + \frac{16952270446279061813}{219902325555200000}}{\mathrm{kd}^2 + 3\mathrm{kd} + \frac{47179}{100}} \tag{8-12}$$

A Tabela 8-9 mostra a inclinação e intercepto relacionado às expressões (8-11) e (8-12), as quais foram obtidas pelo Esquema I.

Tabela 8-9 – Inclinação e intercepto para as expressões do ITAE e tempo de subida exibidas.

Característica	Inclinação	Intercepto
ITAE	0,0001	1,5256e-005
Tempo de subida	7,7478e-005	9,8783e-008

8.1.3 Aplicação do [I]RMOA-II para Sintonia Robusta do Controlador PID no Problema do Motor de Corrente Contínua

O processo de encontrar a melhor configuração dos parâmetros de ganho é chamado de *tuning* ou sintonia, conforme dito na Seção 8.1. Além disso, verificou-se que na literatura os métodos eficientes para realizar a sintonia de controladores PID são formulados como um problema de otimização, o qual tem como objetivo otimizar uma ou mais características da resposta ao degrau unitário e medidas de desempenho relacionadas ao erro no estado estacionário. Diante disso, nesta subseção, apresenta-se o problema multiobjetivo de sintonia com incertezas nos parâmetros de ganho da seguinte forma:

$$\begin{matrix}\min \\ \mathrm{k_p} \in [10^{-10},20] \\ \mathrm{k_i} \in [550,650] \\ \mathrm{k_d} \in [10^{-10},5]\end{matrix} \quad \begin{matrix}\max \\ \mathrm{p} \in [0,0.1]\end{matrix} \quad f_1, f_2 \qquad \text{(8-13)}$$

$$\text{s.a} \qquad g_1, g_2, g_3$$

sendo f_1 o tempo de subida, f_2 o ITAE, g_1 singularidades da análise intervalar (configurações nas quais f_1 ou f_2 retornam *not a number* ou intervalos complexos devem ser descartadas), g_2 outras singularidades (nem o tempo de subida nem o ITAE podem ser menores que zero) e g_3 para verificar a estabilidade do sistema via critério de Routh-Hurwitz e polinômios de Kharitonov conforme explicado em (SOARES, 2008). Escolheram-se os critérios de tempo de subida e ITAE por serem objetivos mais seguros de serem otimizados dado que características como sobre-sinal e tempo de pico podem não existir, uma vez que nem sempre após a aplicação do degrau unitário há sobre-sinal. A planta utilizada é a do problema de posicionamento do motor de corrente contínua cuja função de transferência é definida pela expressão (8-5). Para f_1 e f_2 utilizaram-se as expressões (8-12) e (8-11), respectivamente. Ambas as expressões foram obtidas pelo SNIF-MOGPA conforme mostrado no fim da Subseção 8.1.2. Ressalta-se que a incerteza p foi inserida nas expressões (8-12) e (8-11) sendo somada a toda ocorrência de cada um dos parâmetros de ganho. Portanto, p é uma incerteza paramétrica. Assim, as funções objetivo e as restrições relativas ao problema (8-13) são dadas por:

$$f_1 = \frac{9(\mathrm{kd}+\mathrm{p}) + 2\sin(\mathrm{kp}+\mathrm{p}) + \sqrt{\frac{47179}{100}} + \sqrt{(\mathrm{ki}+\mathrm{p})} + \frac{16952270446279061813}{219902325555200000}}{(\mathrm{kd}+\mathrm{p})^2 + 3(\mathrm{kd}+\mathrm{p}) + \frac{47179}{100}} \qquad \text{(8-14)}$$

$$f_2 = \frac{(\mathrm{kp}+\mathrm{p})}{(\mathrm{kd}+\mathrm{p})^2} + \frac{143907}{10000(\mathrm{kd}+\mathrm{p})} + \frac{(\mathrm{kp}+\mathrm{p})\left((\mathrm{kp}+\mathrm{p}) - \frac{(\mathrm{kp}+\mathrm{p})}{(\mathrm{kd}+\mathrm{p})}\right)}{(\mathrm{kd}+\mathrm{p})^2\left((\mathrm{kd}+\mathrm{p}) + (\mathrm{kp}+\mathrm{p}) - \frac{(\mathrm{kp}+\mathrm{p})}{(\mathrm{ki}+\mathrm{p})}\right)} \quad (8\text{-}15)$$

$$g_1 = \left(isnan(f_1) \vee isnan(f_2) \vee \sim isreal(f_1) \vee \sim isreal(f_2)\right) \text{ for zero.} \quad (8\text{-}16)$$

$$g_2 = f_1 \geq 0 \text{ e } f_2 \geq 0 \quad (8\text{-}17)$$

$$g_3 = (\mathrm{JL})\mathrm{s}^4 + (\mathrm{JR}+\mathrm{BL})\mathrm{s}^3 + (\mathrm{BR}+\mathrm{K}^2+\mathrm{K}(\mathrm{kd}+\mathrm{p})^-)\mathrm{s}^2 + (\mathrm{K}(\mathrm{kp}+\mathrm{p})^+)\mathrm{s} + \mathrm{K}(\mathrm{ki}+\mathrm{p})^+$$
$$\text{e } (\mathrm{JL})\mathrm{s}^4 + (\mathrm{JR}+\mathrm{BL})\mathrm{s}^3 + (\mathrm{BR}+\mathrm{K}^2+\mathrm{K}(\mathrm{kd}+\mathrm{p})^-)\mathrm{s}^2 + (\mathrm{K}(\mathrm{kp}+\mathrm{p})^-)\mathrm{s} + \mathrm{K}(\mathrm{ki}+\mathrm{p})^+ \quad (8\text{-}18)$$
satisfizerem Routh-Hurwitz

sendo que *isnan* e *isreal* são funções do *toolbox* INTLAB (RUMP, 1999). A função *isnan* retorna 1 se o argumento fornecido for *not a number*, caso contrário, retorna 0. A função *isreal* retorna 1 se o argumento fornecido não tiver parte imaginária, caso contrário, retorna 0. Os valores dos parâmetros que aparecem na restrição g_3 foram definidos na Tabela 8-1 localizada na Subseção 8.1.1.

Para resolver (8-13) utilizou-se o algoritmo [I]RMOA II proposto em (SOARES, 2008). O [I]RMOA II tem como objetivo envelopar as soluções robustas eficientes. Inicialmente, divide-se o domínio das incertezas **P** em caixas de acordo com o parâmetro de precisão ε_p, o qual é associado à largura dessas. Em seguida, faz-se o tratamento das restrições que consiste em determinar o subpavimento viável do espaço de busca quando perturbado por todas as caixas obtidas de **P.** O espaço de busca **X** é bisseccionado segundo o parâmetro de precisão ε_x. Com isso, têm-se caixas que ao serem perturbadas: i) satisfazem todas as restrições; ii) não satisfazem nenhuma das restrições; iii) satisfazem algumas restrições. No primeiro caso, a caixa é incluída no subpavimento viável, no segundo pode-se excluir a caixa, e no último é preciso verificar se a caixa ainda não atingiu ε_x e, nesse caso, deve-se bisseccioná-la, caso contrário, como a precisão requerida foi alcançada não se pode afirmar que a caixa é totalmente viável ou inviável. O tratamento das restrições é realizado por testes de inclusão, por exemplo, no caso de uma restrição de desigualdade é verificado se o intervalo de saída da caixa está contido no intervalo $[-\infty, 0]$. Por último, o [I]RMOA II calcula o $\mathbf{u}^{\max}$ para cada caixa do subpavimento viável quando perturbada por todas as caixas obtidas de **P**. Dessa forma, é capaz de realizar a exclusão das caixas não robustas via critério de dominância**.** Ao final da execução do algoritmo as caixas que compõem o envelope sobre as soluções robustas são obtidas. Mais detalhes do [I]RMOA II podem ser consultados em (SOARES, 2008).

Salienta-se que em (SOARES, 2008), visto a dificuldade em obter as funções de inclusão referentes às funções de otimização para um controlador PID, o [I]RMOA II foi aplicado a um problema de sintonia do controlador PD. Assim, nesta tese, uma das contribuições foi tornar o

[I]RMOA II aplicável a problemas com controladores PID. Nesse caso, utilizaram-se funções de inclusão aproximadas, as quais foram obtidas pela aplicação do Teorema 2 às expressões analíticas para as funções de otimização do problema retornadas pelo SNIF-MOGPA.

O resultado da aplicação do algoritmo [I]RMOA II com $\mathbf{u}^{\max}$ para a maximização das incertezas no problema (8-13) é mostrado na Figura 8-3. Os parâmetros $\varepsilon_p = 0{,}05$ e $\varepsilon_x = 0{,}6$ são utilizados no algoritmo. Ressalta-se que alguns pontos da fronteira robusta ao serem impressos no gráfico parecem dominados, porém o que ocorre é que valor de f_2 de tais pontos é bem próximo, diferenciando-se apenas em torno da nona casa decimal.

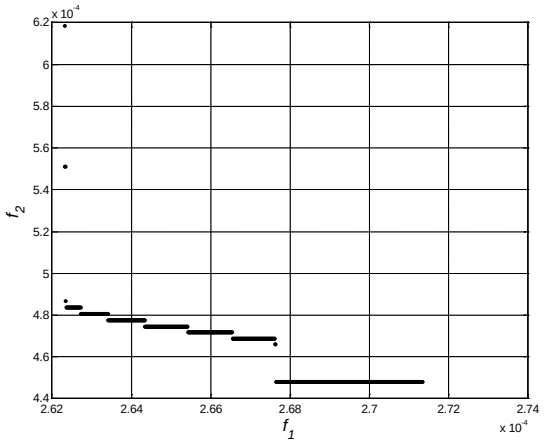


Figura 8-3 – Fronteira robusta do problema (8-13) obtida pelo [I]RMOA-II utilizando-se $\mathbf{u}^{\max}$ com $\varepsilon_x = 0{,}6$ e $\varepsilon_p = 0{,}05$.

A Figura 8-4 exibe o espaço das variáveis de decisão com as 617 soluções robustas encontradas pelo [I]RMOA-II. O parâmetro k_i tem variação considerável dentro da faixa permitida, o k_d tem um valor bem restrito que é de 4,84 em todas as soluções, por fim o k_p apresenta alguma variação dentro do domínio. Assim, nesse problema, o parâmetro k_d é mais sensível ao efeito das incertezas, uma vez que possui um valor único para todas as soluções robustas encontradas. Os valores dos parâmetros de ganho são exibidos como um escalar, pois pega-se o centro da caixa intervalar.

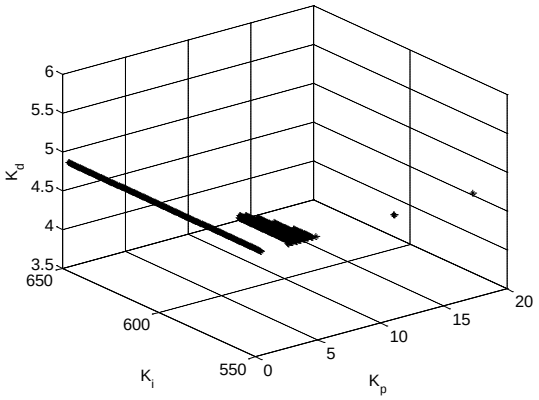


Figura 8-4 – Soluções robustas no espaço das variáveis de decisão.

A Figura 8-5 mostra a resposta ao degrau unitário considerando-se todas as soluções robustas. Nota-se que o controlador PID apresenta boa sintonia para qualquer uma das configurações robustas escolhidas, as quais, inclusive, conduzem a resultados praticamente idênticos.

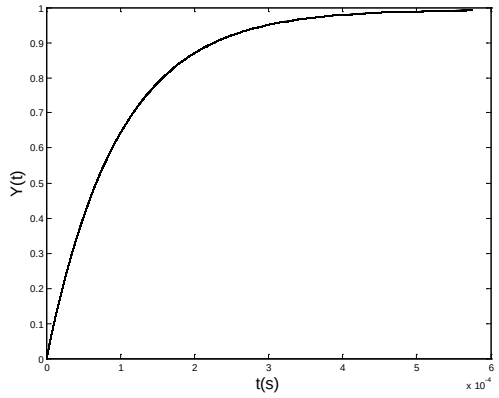


Figura 8-5 – Resposta ao degrau unitário das 617 soluções robustas.

O decisor deve utilizar alguma técnica de tomada de decisão para optar por uma das soluções robustas obtidas. Como ilustração escolheu-se a solução $k_p = 3{,}28$, $k_i = 573{,}63$ e $k_d = 4{,}84$, para a qual se apresentam os valores das funções objetivo considerando-se o pior caso de atuação das incertezas paramétricas: $f_1 = 2{,}66\text{e-}004$ e $f_2 = 4{,}71\text{e-}004$.

8.1.4 Aplicação do SNIF-GPA ao Problema do Controlador PD

Como dito anteriormente, no trabalho de SOARES (2008) são propostos algoritmos para a resolução de problemas de otimização robusta multiobjetivo. Para demonstrar o uso desses algoritmos, Soares os aplica em um problema de sintonia robusta do controlador PD cuja função de transferência $G(s)$ da malha fechada é dada por:

$$G(s) = \frac{k_p + (k_d)s}{s^2 + (1 + k_d)s + k_p}. \tag{8-19}$$

Uma vez que Soares utiliza a análise intervalar para o tratamento das incertezas, faz-se necessário encontrar as expressões analíticas para as características da resposta ao degrau unitário a fim de se obter as respectivas funções de inclusão. Sendo assim, Soares deduziu as expressões necessárias em função dos parâmetros de ganho. No entanto, a tarefa de determinar as expressões analiticamente mostrou-se complexa e demandou considerável manipulação algébrica. Além disso, o procedimento realizado por Soares possui a limitação de ser estendido somente para funções de transferência de ordem menor ou igual a quatro. As expressões obtidas por Soares relativas ao tempo de subida (t_s) e ao ISE para a função de transferência (8-19) foram:

$$t_s = \frac{1}{\sqrt{(k_d + 1)^2 - 4k_p}} \ln\left(\frac{-\sqrt{(k_d + 1)^2 - 4k_p} + 1 - k_d}{\sqrt{(k_d + 1)^2 - 4k_p} + 1 - k_d}\right). \tag{8-20}$$

$$\mathrm{ISE} = -\frac{\left(1+\frac{1-k_d}{\sqrt{(k_d+1)^2-4k_p}}\right)^2}{\sqrt{(k_d+1)^2-4k_p}-1-k_d} - \frac{2\left(1-\left(\frac{1-k_d}{\sqrt{(k_d+1)^2-4k_p}}\right)^2\right)}{-1-k_d} \ldots$$
$$\ldots - \frac{\left(1-\frac{1-k_d}{\sqrt{(k_d+1)^2-4k_p}}\right)^2}{-\sqrt{(k_d+1)^2-4k_p}-1-k_d}. \quad (8\text{-}21)$$

Nota-se que de acordo com o Teorema 2 as funções de inclusão natural obtidas a partir das expressões analíticas tanto do t_s quanto do ISE serão convergentes, monotônicas e estreitas, porém não serão mínimas devido ao problema da dependência. Esse fato acrescido à complexidade matemática e limitação de ordem existente no procedimento realizado por Soares motivou o uso do algoritmo do SNIF-GPA a fim de encontrar aproximações para as Equações (8-20) e (8-21). A partir dessas aproximações aplicando-se o Teorema 2 é possível obter funções de inclusões para o tempo de subida e o ISE. O uso do SNIF-GPA torna o processo de obtenção das funções de inclusão ágil, evita o esforço matemático e não tem limitação imposta pela ordem da função de transferência como há no procedimento de SOARES (2008). A configuração dos parâmetros do SNIF-GPA foi realizada de acordo com os valores da Tabela 8-10. As amostras de treinamento e validação foram obtidas a partir das Equações (8-20) e (8-21), no entanto, simulações computacionais poderiam ter sido utilizadas para esse fim. Para cada característica de resposta ao degrau executou-se o algoritmo dez vezes. Na Tabela 8-11, exibe-se o valor da média, mediana e desvio-padrão (σ) do erro quadrático médio no treinamento e na validação.

Tabela 8-10 – Parâmetros do algoritmo SNIF-GPA.

Parâmetro	**Valor**
Número máximo de gerações	50
Tamanho da população	600
Taxa de cruzamento, mutação e reprodução	0,9, 0,1 e 0,2, respectivamente
Taxa de cruzamento entre nós internos	0,8
Conjunto de funções	+, -, *, / e ^
Conjunto de terminais	k_p, k_d e constante efêmera inteira
Intervalo para constante efêmera inteira	{1, 2,…, 5}
Tamanho do torneio	5
Altura máxima da árvore durante a execução	10
Altura máxima da árvore na geração inicial	4
Método de geração da população inicial	*Ramped half-and-half*
Critério de parada	Número máximo de gerações
Medida de *fitness*	MSE com *fitness scaling*
Conjunto de treinamento - *fitness cases*	100 amostras
Conjunto de validação	900 amostras divididas em 9 grupos

Tabela 8-11 – Média do erro quadrático médio (MSE) para cada característica em dez execuções do SNIF-GPA.

Característica da resposta ao degrau	MSE –Treinamento			MSE – Validação		
	Média	Mediana	σ	Média	Mediana	σ
Tempo de subida	5,83E-03	4,71E-03	3,40E-03	8,37E-02	7,84E-02	4,36E-02
ISE	3,67E-04	7,35E-05	5,19E-04	8,35E-02	5,78E-04	2,53E-01

A seguir, apresentam-se as Equações (8-22) e (8-23), as quais obtiveram os melhores valores de MSE no treinamento para o tempo de subida (1,64E-03) e para o ISE (1,16E-30), respectivamente. Nota-se que a função de inclusão natural obtida a partir da Equação (8-23) será convergente, monotônica, estreita e mínima.

$$t_s = 0{,}322302235892259\ldots \\ \ldots - \frac{-27{,}262578353009}{k_d\left(k_d - k_p + \frac{1}{k_d}\right) - k_d + \left(k_d\left(k_d - k_p + \frac{1}{k_d}\right) - 3\right)\left(k_d - k_p\right) + {k_p}^2 + 24} \tag{8-22}$$

$$\text{ISE} = \frac{\left(2 + \frac{2}{k_p}\right)}{k_d + 1} \tag{8-23}$$

8.1.5 Aplicação do [I]RMOA-II para Sintonia Robusta do Controlador PD

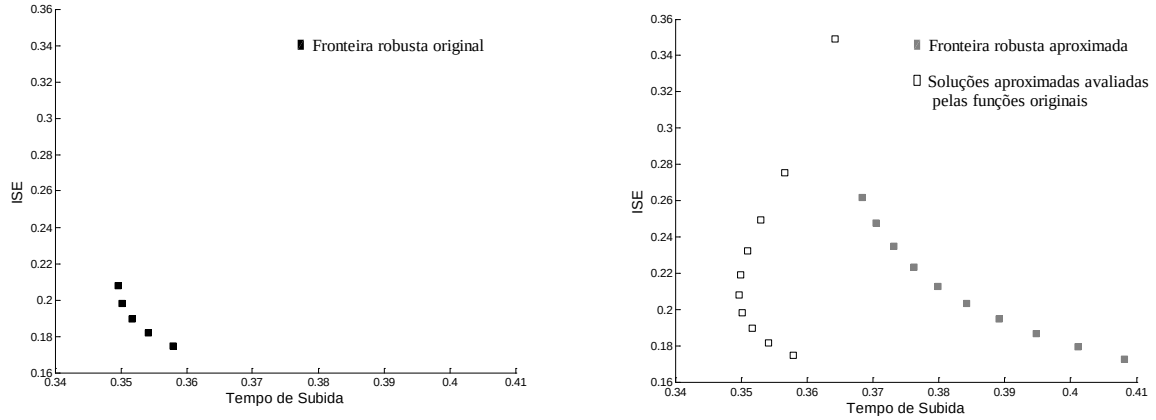
A seguir, resolve-se o problema de sintonia robusta do controlador considerando-se a função de transferência expressa pela Equação (8-19). O tempo de subida e o ISE são os critérios de otimização. Assume-se que a incerteza p está associada aos parâmetros de ganho k_p e k_d da seguinte forma: $k_p + p$ e $k_d + p$, com $p \in [-0{,}1, 0{,}1]$. Para fins de comparação soluciona-se o problema tanto com as Equações originais (8-20) e (8-21) obtidas por (SOARES, 2008) quanto com as Equações aproximadas (8-22) e (8-23) obtidas pelo SNIF-GPA. A formulação do problema é a seguinte:

$$\min_{k_p \text{ e } k_d \in [0{,}15]} \ \max_{p \in [-0{,}1,\ 0{,}1]} \ f_1, f_2 \\ \text{s.a} \quad g_1, g_2, g_3, g_4, \tag{8-24}$$

sendo f_1 o tempo de subida, f_2 o ISE, g_1 e g_2 restrições para evitar que os valores das funções objetivo sejam menores que zero, g_3 para tratar singularidades da matemática intervalar e g_4 para verificar a estabilidade do sistema via critério de Routh-Hurwitz e polinômios de Kharitonov conforme explicado em (SOARES, 2008).

Para resolver (8-24) utilizou-se o [I]RMOA II com $\varepsilon_p = 0{,}05$ e $\varepsilon_x = 0{,}5$. A Figura 8-6 (A) exibe a fronteira robusta gerada pelo [I]RMOA II no espaço dos objetivos quando se utilizam as funções objetivo originais. A Figura 8-6 (B) também mostra o espaço dos objetivos, sendo que os quadrados preenchidos na cor cinza representam a fronteira robusta quando se utilizam as funções objetivo aproximadas. Nota-se que, apesar da proximidade da escala, o uso das funções aproximadas fez com que o algoritmo retornasse mais soluções robustas. Provavelmente, o erro de aproximação existente

nas funções aproximadas, principalmente na função relativa ao tempo de subida, contribuiu para que as soluções excedentes fossem encontradas. Diante disso, fez-se o seguinte: para cada ponto da fronteira robusta obtida com as funções aproximadas verificou-se a respectiva solução no espaço das variáveis de decisão. Observou-se que dentre as dez soluções encontradas, cinco eram idênticas às soluções obtidas com o uso das funções originais, o que é mostrado na Figura 8-7, na qual os dez quadrados representam as soluções robustas com as funções aproximadas e os cinco pontos pretos as soluções robustas com as funções originais. Diante disso, avaliaram-se todas as dez soluções obtidas com o uso das funções aproximadas utilizando-se as funções originais, o resultado é apresentado na Figura 8-6 (B) pelos quadrados brancos. Nota-se que dentre esses pontos, apenas cinco são não-dominados e, além disso, que esses cinco pontos não-dominados são idênticos aos pontos da fronteira robusta da Figura 8-6 (A).



(A) (B)

Figura 8-6 – Espaço dos objetivos do problema (8-24).

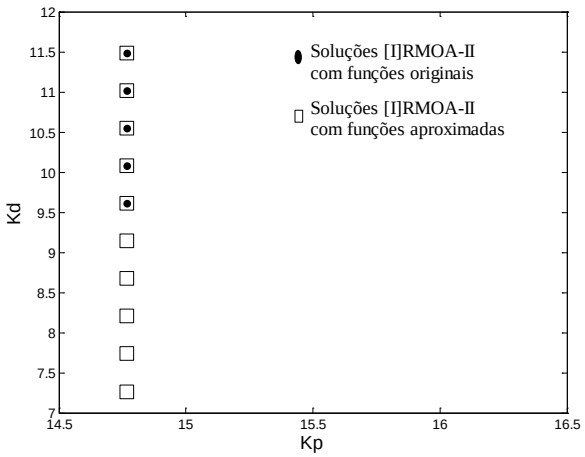
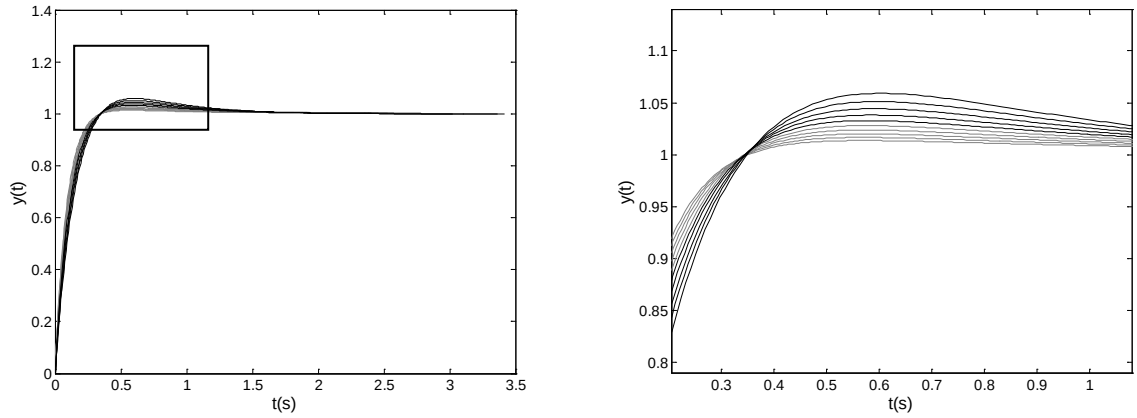


Figura 8-7 – Espaço das variáveis de decisão do problema (8-24).

Em relação ao tempo computacional, o [I]RMOA II com o uso das funções aproximadas gastou 35,1% do tempo consumido com a utilização das funções originais.

A Figura 8-8 exibe a resposta ao degrau unitário para as dez soluções robustas da Figura 8-7. Em cinza, tem-se o resultado para as cinco soluções robustas em comum obtidas tanto com o uso das funções originais quanto com as funções aproximadas no [I]RMOA II e, em preto, tem-se a resposta para as cincos soluções excedentes encontradas devido ao uso das funções aproximadas.



(A) (B)

Figura 8-8 – Resposta do sistema ao degrau unitário de acordo com as soluções da Figura 8-6 (B).

Diante do exposto, conclui-se que o processo de otimização com o uso das funções aproximadas fez com que o algoritmo robusto economizasse tempo considerável, provavelmente, devido ao fato da expressão aproximada para o ISE resultar em uma inclusão mínima. Além disso, verificou-se que é importante utilizar as funções de otimização originais ou simulação computacional, quando for o caso, para realizar a avaliação final das soluções resultantes do processo de otimização realizado via funções aproximadas.

8.2 Aproximação de Funções de Inclusão em Eletromagnetismo

A sociedade internacional COMPUMAG[10] sugere um rol de problemas de *benchmark* intitulados TEAM (*Testing Electromagnetic Analysis Methods*). Dentre esses, tem-se o problema TEAM 22 (ALOTTO *et al.*, 2008) para o qual SOARES *et al.* (2009a) propõem uma versão robusta multiobjetivo que considera incerteza nas variáveis de decisão. Para resolver essa versão robusta, SOARES *et al.* (2009a) empregaram um algoritmo genético multiobjetivo. Entretanto, pelo fato de não terem em mãos a versão analítica das funções de otimização, uma vez que os valores dessas são calculados por simulações realizadas pelo método dos elementos finitos, não puderam aplicar algoritmos intervalares. Dessa forma, optaram por lidar com as incertezas utilizando-se de uma abordagem probabilística. Nesta seção, objetiva-se aplicar o SNIF-GPA para aproximar inclusões para

[10] http://www.compumag.org

as funções de otimização do problema TEAM 22 e, em seguida, resolver sua versão robusta utilizando o IRMOEA-M.

O problema TEAM 22 robusto possui a configuração exibida na Figura 8-9. Os parâmetros $R_1, h_1, e\, d_1$ são conhecidos e os parâmetros de projeto $R_2, h_2, e\, d_2$ devem ser otimizados considerando-se o pior caso dos parâmetros de incerteza $p_1, p_2, e\, p_3$, os quais representam, por exemplo, erros de medida nas variáveis de projeto (SOARES *et al.*, 2009a). A ideia geral é a concepção de um dispositivo capaz de armazenar energia por meio de campos magnéticos. O sistema é composto de duas bobinas concêntricas com densidade de corrente em sentidos opostos. A primeira bobina é responsável por armazenar a energia e a segunda deve ser projetada para diminuir a dispersão magnética causada pela primeira bobina (ALOTTO *et al.*, 1996b; ALOTTO *et al.*, 2008).

A formulação matemática do problema TEAM 22 com incertezas é dada em (SOARES *et al.*, 2009a). Trata-se de um problema de minimização com duas funções objetivo e uma restrição. A primeira função objetivo (F_1) computa o campo de dispersão, que é avaliado ao longo de 22 pontos nas linhas a e b, conforme mostrado na Figura 8-9. A segunda função objetivo (F_2) mede o desvio percentual da energia computada em relação ao valor de 180 MJ (valor de referência que se deseja armazenar). A restrição C_1 assegura que o campo magnético não viole a condição física que garante supercondutividade.

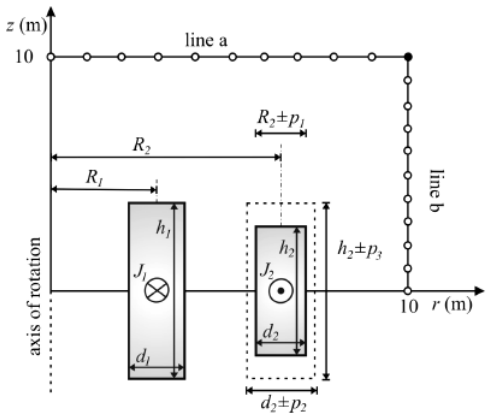


Figura 8-9 – Problema TEAM 22 com incertezas (SOARES *et al.*, 2009a).

8.2.1 Aplicação do SNIF-GPA ao Problema TEAM 22

Nesta subseção, mostra-se a aplicação do SNIF-GPA ao problema TEAM 22 a fim de se obter as funções de inclusões para as funções do problema de otimização. Dessa forma, objetiva-se encontrar aproximações analíticas para F_1, F_2 e C_2 em termos dos parâmetros de otimização $R_2, h_2, e\, d_2$ para que a partir do Teorema 2 sejam obtidas as funções de inclusão. A Figura 8-10 exibe a metodologia proposta. As amostras para o conjunto de treinamento e validação foram obtidas via simulações realizadas utilizando-se o Método dos Elementos Finitos (FEM - *Finite Element Method*) configurado com elementos triangulares, primeira ordem e fator de malha igual a 0,75. Foram feitas 450 simulações com valores escolhidos aleatoriamente para $R_2, h_2, e\, d_2$, respeitando-se o domínio

permitido para cada um desses, que é $[2{,}6, 3{,}4]$, $[0{,}408, 2{,}2]$ e $[0{,}1, 0{,}4]$, respectivamente. Estabeleceram-se o conjunto de treinamento com 360 amostras e o conjunto de validação com 90 amostras.

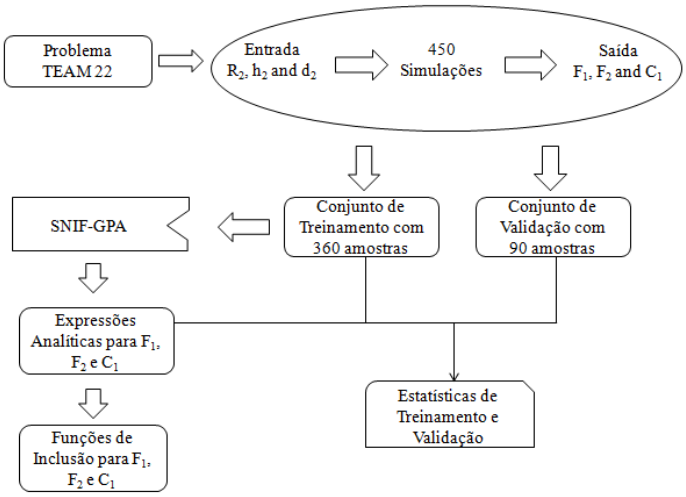


Figura 8-10 – Metodologia proposta para aplicação do SNIF-GPA ao problema TEAM 22.

A fim de verificar a confiabilidade das aproximações obtidas, utilizou-se o Wilcoxon *rank sum test*, o qual indica se existe diferença evidente entre dois conjuntos de dados. Para a computação do teste utilizou-se a função *ranksum* do Matlab® com um nível de confiança de 95% para comparar as saídas aproximadas (dada pelas expressões obtidas) com as saídas esperadas (calculas via FEM). Se o p-valor do teste é maior ou igual a 0,05, o resultado do teste indica que não há diferença significativa entre os dois conjuntos de dados para o nível de confiança especificado.

O SNIF-GPA foi executado dez vezes conforme a configuração listada na Tabela 8-12. Os resultados do treinamento e da validação são apresentados nas Tabelas 8-13, 8-14 e 8-15. As execuções que retornam os melhores indivíduos (menor MSE no treinamento) são destacadas com sombreado cinza. Verifica-se que as aproximações retornadas em todas as execuções são estatisticamente válidas de acordo com o Wilcoxon *rank sum test*. Além disso, possuem coeficiente de determinação R2 bem próximo de 1.

Tabela 8-12 – Configuração do SNIF-GPA.

Parâmetro	Valor
Número de gerações	100
Número de programas	500
Probabilidades de cruzamento, mutação e elitismo.	0,9, 0,3 e 0,2, respectivamente
Conjunto de funções	+, -, *, /, ^, √, log, seno e exp
Conjunto de terminais	R_2, h_2, d_2 e constante efêmera
Tamanho do torneio	5
Altura máxima da árvore:	
- na geração inicial	2
- durante a execução	18
Método de geração da população inicial	*Ramped half-and-half*

Tabela 8-13 – Resultados do SNIF-GPA para F_1.

Execução	Valores obtidos para F_1					
	Treinamento			Validação		
	MSE	R2	p-valor	MSE	R2	p-valor
1	2,15	0,999	0,593	2,596	0,999	0,648
2	9,429	0,997	0,617	10,183	0,997	0,776
3	11,34	0,997	0,278	11,025	0,996	0,878
4	4,65	0,999	0,423	4,908	0,998	0,584
5	10,59	0,997	0,405	10,91	0,996	0,930
6	6,687	0,998	0,294	8,022	0,997	0,614
7	9,325	0,997	0,366	9,505	0,997	0,940
8	14,46	0,996	0,241	14,423	0,995	0,513
9	8,359	0,998	0,350	8,811	0,997	0,851
10	5,632	0,998	0,453	6,785	0,998	0,488
Média	8,262	0,998	0,402	8,716	0,997	0,722
Desvio-padrão	3,584	0,001	0,126	3,360	0,001	0,173

Tabela 8-14 – Resultados do SNIF-GPA para F_2.

Execução	Valores obtidos para F_2					
	Treinamento			Validação		
	MSE	R2	p-valor	MSE	R2	p-valor
1	0,006	0,991	0,529	0,006	0,989	0,412
2	0,015	0,977	0,161	0,014	0,976	0,169
3	0,007	0,990	0,512	0,009	0,985	0,519
4	0,011	0,983	0,264	0,013	0,978	0,239
5	0,014	0,979	0,462	0,015	0,975	0,125
6	0,012	0,982	0,519	0,014	0,977	0,358
7	0,011	0,984	0,13	0,012	0,979	0,128
8	0,014	0,978	0,215	0,013	0,978	0,217
9	0,010	0,984	0,553	0,011	0,981	0,218
10	0,013	0,980	0,349	0,012	0,980	0,221
Média	0,011	0,983	0,369	0,012	0,980	0,261
Desvio-padrão	0,003	0,005	0,166	0,003	0,004	0,129

TABELA 8-15 – RESULTADOS DO SNIF-GPA PARA C_1.

Execução	Valores obtidos para C_1					
	Treinamento			Validação		
	MSE	R2	p-valor	MSE	R2	p-valor
1	0,086	0,940	0,859	0,087	0,935	0,778
2	0,091	0,937	0,866	0,091	0,932	0,802
3	0,086	0,940	0,852	0,079	0,941	0,754
4	0,088	0,939	0,890	0,087	0,935	0,833
5	0,080	0,945	0,797	0,060	0,955	0,798
6	0,085	0,941	0,838	0,080	0,938	0,722

Execução	Valores obtidos para C_1					
	Treinamento			Validação		
	MSE	R2	p-valor	MSE	R2	p-valor
7	0,073	0,949	0,759	0,065	0,951	0,737
8	0,091	0,937	0,902	0,090	0,932	0,825
9	0,085	0,941	0,841	0,085	0,936	0,752
10	0,082	0,943	0,722	0,077	0,942	0,656
Média	0,085	0,941	0,833	0,080	0,940	0,766
Desvio-padrão	0,005	0,004	0,057	0,010	0,008	0,053

8.2.2 Análise Comparativa do SNIF-GPA no Problema TEAM 22 com Incertezas

Nesta subseção, os resultados obtidos pelo SNIF-GPA são comparados com os resultados de outras duas técnicas de obtenção de modelos aproximados, são elas: Redes Neurais com Funções de Base Radial (RBF-NN - *Radial Basis Function Neural Networks*) e *Universal* Kriging (MENDES *et al*., 2012a; MENDES *et al*., 2012b). A comparação dos resultados foi realizada com base em dois critérios: i) qualidade das expressões aproximadas, medida em termos do MSE, e ii) adequação à otimização robusta intervalar, que é medida pela largura do intervalo da saída da função de inclusão obtida. A importância do segundo critério está relacionada ao fato de que, quanto mais estreito é o intervalo de saída, mais precisos serão os resultados e, além disso, mais rápida será a avaliação da função de inclusão pelos algoritmos intervalares.

As RBF-NNs foram introduzidas por (BROOMHEAD e LOWE, 1988) e, têm vantagens em relação a outros tipos de redes neurais, por exemplo, melhor capacidade de aproximação, estrutura mais simples e algoritmos de aprendizado mais rápidos. Uma RBF-NN é composta por três camadas: a de entrada, a escondida e a de saída. A camada de entrada é constituída de nós sensoriais. A camada escondida é formada por neurônios de funções de base radial. A camada de saída é linear e fornece a resposta da rede. A função de ativação da camada escondida é normalmente a função Gaussiana e computa a distância Euclidiana entre o sinal de entrada e um vetor parâmetro da rede. Mais detalhes sobre RBF-NN podem ser obtidos em (BRAGA *et al.*, 2007). A RBF-NN utilizada nesta seção é gerada pela função *newrb* disponível no ANN *Toolbox* do Matlab®. A função *newrb* cria uma RBF-NN adicionando um neurônio por vez. Os neurônios são adicionados até que se atinja a meta de erro ou que o número máximo de neurônios seja alcançado (BEALE *et al.*, 2011). Para a obtenção das aproximações utilizaram-se 100 neurônios na camada escondida, o restante dos parâmetros foi estabelecido pelos valores *default* da função *newrb.*

Universal Kriging usa dois conjuntos de funções base. O primeiro conjunto objetiva seguir a tendência geral da função a ser modelada e, geralmente, é composto de polinômios $p(x)$. O segundo conjunto torna possível seguir as variações ao redor da tendência geral $h(x)$ e, tradicionalmente, é composto por um conjunto de funções de base radial Gaussianas ou multiquadráticas. O resultado

final é uma combinação linear de funções. No entanto, ressalta-se que os dois conjuntos de funções desempenham diferentes papéis, que são distinguidos na seguinte expressão (LEBENSZTAJN *et al.*, 2004):

$$\tilde{y}(\mathbf{x}) = \mathbf{w}^T\mathbf{h}(\mathbf{x}) + \mathbf{c}^T\mathbf{p}(\mathbf{x}). \tag{8-25}$$

Nesta expressão, existem N + m incógnitas: os coeficientes $w_1, w_2, \ldots, w_N$ e $c_1, c_2, \ldots, c_m$. A fim de se determinar essas incógnitas procede-se em dois estágios: primeiro, assume-se que $c_1, c_2, \ldots, c_m$ são conhecidos e determina-se os valores de $w_1, w_2, \ldots, w_N$ de forma que a combinação linear passe pelos N pontos de medida e, em seguida, ajustam-se $c_1, c_2, \ldots, c_m$ (preservando a passagem pelos pontos de medida) tais que a variação $\mathbf{w}^T\mathbf{h}(\mathbf{x})$ em torno de $\mathbf{c}^T\mathbf{p}(\mathbf{x})$ seja a menor possível a fim de se obter a aproximação mais suave permitida.

A partir de (8-25) reescrita sob a forma matricial

$$y = \mathbf{H}\,\mathbf{w} + \mathbf{P}\,\mathbf{c}\,, \tag{8-26}$$

obtêm-se

$$\mathbf{c} = [[\mathbf{H}^{-1}\mathbf{P}]^T\mathbf{P}]^{-1}[\mathbf{H}^{-1}\mathbf{P}]^T\mathbf{y} \quad \text{e}$$
$$\mathbf{w} = \mathbf{H}^{-1}[\mathbf{y} - \mathbf{P}\,\mathbf{c}]. \tag{8-27}$$

O modelo de *Universal* Kriging desta subseção utilizou polinômios de segunda ordem para p(x) e funções de base radial multiquadráticas para h(x), como explicado em (ALOTTO *et al.*, 1996a).

As amostras utilizadas para obtenção das aproximações foram as mesmas utilizadas na Subseção 8.2.1. Os resultados com relação ao critério de qualidade das expressões aproximadas para a RBF-NN e *Universal* Kriging são apresentados nas tabelas 8-16 e 8-17, respectivamente. Verifica-se que as aproximações retornadas tanto pela RBF-NN quanto pelo *Universal* Kriging são válidas de acordo com o Wilcoxon *rank sum test* com nível de confiança de 95% (p-valor $\geq$ 0,05). Além disso, possuem coeficiente de determinação R2 bem próximo de 1. Observa-se que os modelos aproximados obtidos pelo *Universal* Kriging possuem valores de MSE no treinamento bem próximos de zero. Isso se deve à característica que essa técnica possui de fazer com que o modelo obtido passe por todos os pontos de treinamento. Nota-se, também, que a ordem de magnitude do MSE na validação para o SNIF-GPA, RBF-NN e *Universal* Kriging é a mesma considerando-se as aproximações geradas para F_1, F_2 e C_1.

TABELA 8-16 – RESULTADO DA RBF-NN PARA O PROBLEMA TEAM 22.

	Valores Obtidos Usando RBF-NN					
	Treinamento			**Validação**		
	MSE	**R2**	**p-valor**	**MSE**	**R2**	**p-valor**
F_1	1,061	0,999	0,952	2,306	0,999	0,890
F_2	0,001	0,999	0,961	0,002	0,997	0,772
C_1	0,020	0,985	0,896	0,052	0,961	0,820

TABELA 8-17 – RESULTADO DO *UNIVERSAL* KRIGING PARA O PROBLEMA TEAM 22.

	Valores Obtidos Usando *Universal* Kriging					
	Treinamento			**Validação**		
	MSE	**R2**	**p-valor**	**MSE**	**R2**	**p-valor**
F_1	1E-23	1	0,996	6,464	0,997	0,919
F_2	1E-27	1	0,988	0,002	0,995	0,704
C_1	1E-26	1	0,997	0,047	0,964	0,992

Na Tabela 8-18 apresenta-se o resultado com respeito ao critério de adequação à otimização robusta intervalar. Os resultados foram obtidos fazendo-se a avaliação intervalar de cada uma das funções de inclusões aproximadas no domínio permitido para $R_2, h_2, e\, d_2$ conforme descrito na subseção anterior. Ressalta-se que para o SNIF-GPA, o resultado apresentado é a mediana da largura do intervalo de saída considerando-se todas as dez execuções. Isso foi feito uma vez que o valor de largura de saída das dez expressões obtidas para cada função de otimização F_1, F_2 e C_1 não segue uma distribuição normal de acordo com o teste estatístico Lilliefors. O valor de referência foi computado pela diferença entre os valores máximo e mínimo dos intervalos de saída de F_1, F_2 e C_1, considerando-se as amostras de treinamento e validação. Os resultados mostram que as inclusões aproximadas obtidas pelo SNIF-GPA possuem intervalos de saída significativamente mais estreitos do que as inclusões obtidas pelas outras duas técnicas. Dessa forma, as inclusões geradas pelo SNIF-GPA mostram-se mais adequadas para serem utilizadas em métodos intervalares. Estes resultados não invalidam o uso das inclusões obtidas pelos outros métodos, pois há procedimentos capazes de lidar com inclusões mais largas. Entretanto, esses procedimentos, em geral, consomem tempo computacional considerável e, portanto, devem ser evitados.

TABELA 8-18 – RESULTADO PARA O CRITÉRIO DE ADEQUAÇÃO À OTIMIZAÇÃO ROBUSTA INTERVALAR.

	Largura do Intervalo de Saída para:		
	F_1	**F_2**	**C_1**
Referência	447,9533	5,3504	5,5030
SNIF-GPA	1,38E+03	71,1565	27,6316
RBF-NN	1,76E+11	1,3E+10	5,19E+10
***Universal* Kriging**	1,08E+05	2,36E+03	1,05E+04

O desenvolvimento do trabalho realizado nesta subseção e na subseção anterior resultou em um artigo intitulado "*Appraisal of Surrogate Modeling Techniques: A Case Study of Electromagnetic Device*", o qual foi aprovado para publicação no periódico IEEE Transactions on Magnetics.

8.2.3 Aplicação do IRMOEA-M ao Problema TEAM 22 com Incertezas

Como dito anteriormente, SOARES *et al.* (2009a) não puderam aplicar algoritmos com tratamento intervalar da incertezas na versão robusta do Problema TEAM 22 devido à indisponibilidade das inclusões para as funções de otimização. Nesta subseção, utilizam-se as inclusões aproximadas para

F_1, F_2 e C_1 obtidas com o SNIF-GPA no IRMOEA-M. Além disso, compara-se o resultado dessa metodologia àquele alcançado pela abordagem probabilística proposta em (SOARES *et al.*, 2009a).

Na abordagem proposta em (SOARES *et al.*, 2009a) utiliza-se um algoritmo evolucionário multiobjetivo denominado MOGA. Esse algoritmo possui codificação binária, cruzamento com um ponto de corte, mutação baseada na troca de bit, seleção pela técnica da roleta, utiliza técnica de nicho e é elitista. A população é constituída de 50 indivíduos e o algoritmo é executado por 50 gerações. Para lidar com os parâmetros de incerteza, utiliza-se perturbar cada indivíduo por um determinado número finito de amostras de incertezas (5, 10 ou 20) considerando-se valores admissíveis dos parâmetros de incerteza que são 0,03, 0,01 e 0,01 para $p_1, p_2, \text{e } p_3$, respectivamente. Essas amostras são utilizadas para computar a chamada aproximação do cenário de pior caso (WCSA – *Worst Case Scenario Approximation*) de cada indivíduo, que, resumidamente, consiste em calcular o ponto ideal de maximização dado o conjunto de amostras e utilizá-lo como ponto de referência durante o processo de otimização. No trabalho, foi mostrado que quanto maior o número de amostras, melhores valores para o pior caso são retornados. Assim, optou-se comparar a abordagem utilizando-se 20 amostras de incerteza.

No MOGA com WCSA utilizam-se exclusivamente simulações pelo FEM para o cômputo dos valores das funções de otimização. Nesta subseção, a proposta é utilizar o IRMOEA-M com as inclusões aproximadas durante o processo de otimização e somente após a última geração avaliar as soluções obtidas pelo FEM (nesse caso, utilizando-se as 20 amostras de incerteza e o WCSA). Além disso, o IRMOEA-M é executado n_t vezes. As soluções não-dominadas são armazenadas em um arquivo, o qual é atualizado ao fim de cada execução, sendo as soluções dominadas removidas. Finalizadas as n_t execuções, tem-se que a fronteira robusta é constituída pelas soluções existentes no arquivo. Os experimentos foram realizados para $n_t = 5$. O IRMOEA-M foi configurado com 30 indivíduos, 20 gerações, $\varepsilon_p = 0{,}02$, probabilidade de cruzamento de 90% e mutação de 10%. A seguir, são exibidas as aproximações utilizadas para as funções de otimização. Para melhor apresentação, as expressões foram simplificadas utilizando-se a função *simplify* do Matlab®.

$$F_1 = -\frac{1893\left(d_2\, r_2 + \frac{1231\, d_2}{500} + \frac{d_2}{r_2}\right)\sin(h_2)}{625} + \frac{1893\, d_2\, {r_2}^2\, {h_2}^{\frac{5}{2}}\left(d_2\, {r_2}^2 - \frac{r_2}{h_2}\right)}{625} + \frac{1893\, d_2\, {h_2}^2\, {r_2}^2\,(d_2\, {r_2}^2 - 1)}{625} - \frac{2330283\sqrt{h_2}\, d_2\, r_2}{312500} - \frac{2330283\, d_2\, r_2}{312500} - \frac{2868578373\, d_2\, h_2}{156250000} + \frac{38377}{2000} \tag{8-28}$$

$$F_2 = -d_2\, h_2\, {r_2}^3 \cdot (1{,}0624)^{h_2}\, \sin(d_2\, r_2 + d_2)\, \sin\left({h_2}^{d_2} + d_2 + r_2\right)\left(\frac{50773\, {h_2}^{d_2}}{1000000} + \frac{50773\, d_2}{500000}\right) \sin\left(\sin\left(d_2\left({h_2}^{d_2} + r_2\right)\right)\right) \sin\left(\sin\left(\sin\left((d_2 + h_2 + r_2)\sin\left({h_2}^{d_2} + d_2 + r_2\right) + d_2\right)\right)\right) \log\left(d_2\, r_2\, {h_2}^2\,(d_2\, r_2 + d_2)\right) + 0{,}1596 \tag{8-29}$$

$$C_1 = \frac{\text{NUM}}{625\, r_2} + 3{,}4144, \tag{8-30}$$

com

$$\text{NUM} = 4058\, d_2 \left(sin\left(sin\left(d_2\, h_2 + \frac{200\, d_2}{269} + \frac{h_2}{r_2} \right) sin\left(d_2\, h_2 \left(d_2\, h_2 - sin(1)\, sin\left(\frac{d_2}{r_2}\right) {r_2}^2 \right) \right) + sin\left(sin\left(d_2\, h_2 + \frac{h_2}{r_2} \right) \right) sin\left(e\, h_2\, p\, x\, {d_2}^2 \right) \right) + \right.$$

$$sin\left(sin\left(d_2\, h_2 + \frac{200\, d_2}{269} + \frac{h_2}{r_2} \right) sin\left(d_2\, h_2 \left(d_2\, h_2 - sin\left(\frac{d_2}{h_2}\right) sin\left(\frac{d_2}{r_2}\right) {r_2}^2 \right) \right) + d_2\, sin(r_2) \right) +$$

$$sin\left(sin\left({d_2}^2 \right) sin\left(d_2\, h_2 \left(d_2\, h_2 - sin\left(\frac{d_2}{h_2}\right) sin\left(\frac{d_2}{r_2}\right) {r_2}^2 \right) \right) + d_2\, sin(r_2) \right) +$$

$$sin\left(\frac{sin\left(sin\left(d_2\, h_2 + \frac{d_2}{r_2} \right) + sin\left({h_2}^2 + \frac{h_2}{r_2} \right) \right)}{r_2} + d_2\, sin(r_2) + e^2\, {d_2}^2\, p^2\, x^2 \left(d_2\, h_2 + sin\left(\frac{d_2}{r_2}\right) d_2\, h_2 \right) \right) + sin\left(\frac{sin\left(sin\left(d_2\, h_2 + \frac{d_2}{r_2} \right) + sin\left({h_2}^2 + \frac{d_2}{r_2} \right) \right)}{r_2} + d_2\, sin(r_2) + \right.$$

$$\left. e^2\, {d_2}^2\, p^2\, x^2 \left(\frac{h_2\, {d_2}^2}{r_2} + d_2\, h_2 \right) \right) + sin\left(sin\left(log\left(d_2\, h_2 + \frac{d_2}{r_2} \right) \right) sin\left(d_2\, h_2 + \frac{200\, d_2}{269} + \frac{h_2}{r_2} \right) + d_2\, sin(r_2) \right) + sin\left(\frac{sin\left(sin(d_2\, h_2) + sin\left({h_2}^2 + \frac{d_2}{r_2} \right) \right)}{r_2} + \right.$$

$$\left. \left. e^2\, {d_2}^2\, p^2\, x^2 \left(d_2\, h_2 + sin\left(\frac{d_2}{r_2}\right) d_2\, h_2 \right) + \frac{sin\left(\frac{d_2}{r_2}\right)}{r_2} \right) + sin\left(e\, h_2\, p\, x\, {d_2}^2\, sin\left(d_2\, h_2 + \frac{200\, d_2}{269} + \frac{h_2}{r_2} \right) + \frac{log\left(d_2\, h_2 + \frac{d_2}{r_2} \right)}{r_2} \right) \right)$$

Na Figura 8-11 comparam-se as fronteiras robustas obtidas pelo MOGA com WCSA e pelo IRMOEA-M. Os resultados indicam que as fronteiras obtidas estão bem próximas.

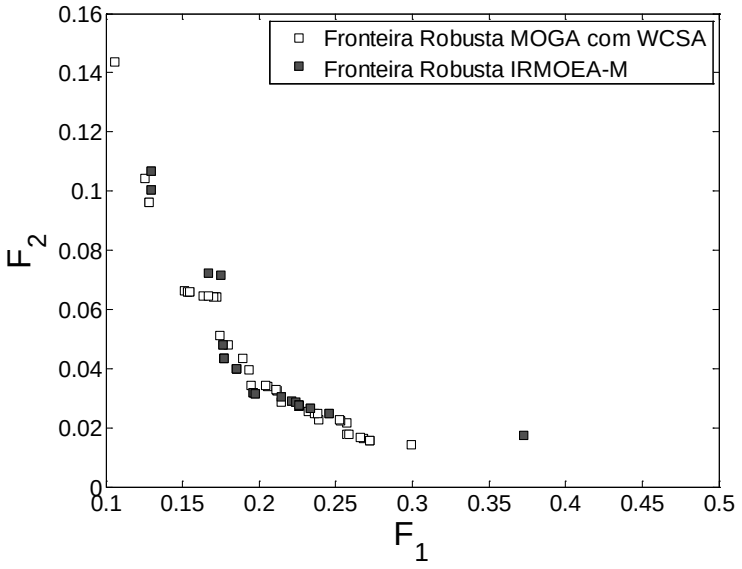


Figura 8-11 – Comparação entre as fronteiras robustas.

A partir das soluções existentes na fronteira robusta obtida é necessário selecionar uma solução robusta como resposta para o problema TEAM 22. Em (GUIMARÃES *et al.*, 2006b) e (SOARES *et al.*, 2009a) essa seleção foi realizada descartando-se as soluções com densidade de fluxo magnético *Bstray* maior que 3 e selecionando-se a solução que conduz ao menor volume do material $V_{min} = 2\pi R_2 h_2 d_2$. Sendo assim, dada a fronteira robusta obtida pelo IRMOEA-M, selecionou-se

$$[R_2\ h_2\ d_2] = [3{,}3823 \quad 0{,}4858 \quad 0{,}3184], \tag{8-31}$$

resultando na energia armazenada de 180,42 MJ, $Bstray = 1{,}93$ mT e $V_{min} = 3{,}287\ m^3$. Para efeitos comparativos, a solução escolhida em (SOARES *et al.*, 2009a) foi

$$[R_2\ h_2\ d_2] = [3{,}3790 \quad 0{,}4887 \quad 0{,}3168], \tag{8-32}$$

resultando na energia armazenada de 180,36 MJ, $Bstray = 1{,}38$ mT e $V_{min} = 3{,}287\ m^3$.

Um fator relevante a ser analisado é o custo computacional, que inclui os custos: i) de geração das amostras, ii) da construção do modelo aproximado e iii) do processo de otimização. Dessa forma, compara-se o custo computacional total do algoritmo MOGA com WCSA em relação ao da abordagem SNIF-GPA + IRMOEA-M.

O primeiro custo envolve as 450 simulações FEM necessárias para alimentar o SNIF-GPA. Cada avaliação FEM gasta 3,5 segundos em média, assim, esse custo é estimado em 1575s.

O segundo custo é específico do SNIF-GPA, o qual gasta em média 2700 segundos para obter a inclusão para cada função de otimização. Assim, esse custo é estimado em 8100s.

O terceiro custo envolve o MOGA com WCSA e o IRMOEA-M. No caso do MOGA com WCSA o custo é dado pela quantidade de simulações FEM necessárias, que é de 50000 $(50 \times 20 \times 50)$, i.e., 50 indivíduos, 20 amostras de incertezas e 50 gerações. Consequentemente, o custo estimado é de 175000s. No caso do IRMOEA-M, como cada execução gasta 1850s e são realizadas 5, tem-se 9250s. Além disso, a última geração é avaliada via FEM com 20 amostras de incerteza. Sendo assim, acrescenta-se 3000 $(30 \times 20 \times 5)$ simulações FEM, i.e., 30 indivíduos, 20 amostras de incertezas e 5 execuções, que resulta em 10500s. Dessa forma, o terceiro custo para o IRMOEA-M é estimado em 19750s.

O custo total do MOGA com WCSA é estimado em 175000s, uma vez que o primeiro e o segundo custos não são aplicáveis nesse caso. Um total de 50000 simulações FEM é requerido.

O custo total da abordagem SNIF-GPA + IRMOEA-M é estimado em 29425s (1575s + 8100s + 19750s). Em termos de simulações FEM, 3450 são necessárias.

Portanto, a abordagem SNIF-GPA + IRMOEA-M requer cerca de 16,8% do tempo de CPU e 6,9% de simulações FEM em relação ao que é requerido pelo MOGA com WCSA. A Tabela 8-19 sumariza os resultados. Um computador com CPU i3 e 4 GB de memória RAM foi utilizado nos experimentos.

Tabela 8-19 – Comparação do custo computacional.

	MOGA com WCSA		**SNIF-GPA + IRMOEA-M**	
Custo	**Tempo Estimado (s)**	**Número de Simulações FEM**	**Tempo Estimado (s)**	**Número de Simulações FEM**
1º	Não aplicável	Não aplicável	1575	450
2º	Não aplicável	Não aplicável	8100	0
3º	175000	50000	19750	3000
Total	175000	50000	29425	3450

A Figura 8-12 ilustra a importância das soluções robustas para o problema TEAM 22 com incertezas. Os quadrados representam a fronteira não-dominada relativa às soluções nominais obtidas

pelo MOGA, isto é, quando não são consideradas as incertezas paramétricas no problema. Os círculos indicam a fronteira robusta obtida pelo MOGA com WCSA, utilizando-se 20 amostras de incerteza. Os pontos pretos mostram o que ocorre com as soluções nominais quando essas são perturbadas utilizando-se 20 amostras de incerteza. Verifica-se que o desempenho de muitas soluções nominais, quando sujeitas às incertezas, é comprometido e torna-se visivelmente pior que o desempenho das soluções robustas.

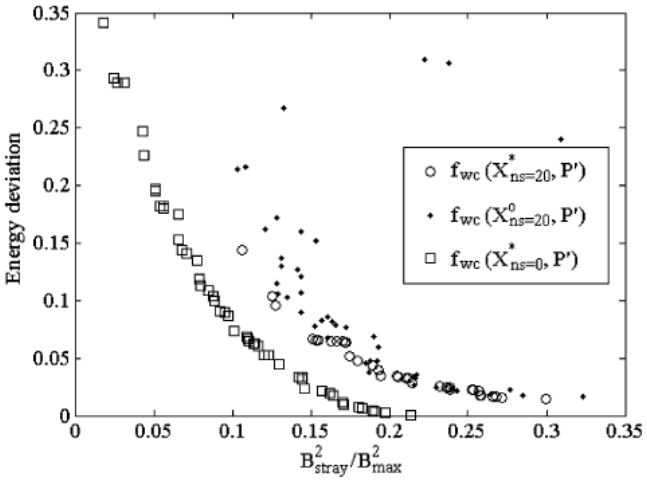


Figura 8-12 – Importância das soluções robustas no problema TEAM 22, extraído de (SOARES *et al*., 2009a).

As ideias desenvolvidas nesta subseção e na Subseção 8.2.1 resultaram em um artigo intitulado "*A Surrogate Genetic Programming Based Model to Facilitate Robust Multi-Objective Optimization: A Case Study in Magnetostatics*" aceito para publicação no periódico IEEE Transactions on Magnetics.

8.3 Aplicação do SNIF-MOGPA ao Problema Desafio

Em novembro de 2008, Neumaier postou na lista de e-mails de pesquisadores em IA um problema, intitulado problema desafio, relacionado à obtenção de um envelope[11] para uma função real. Estudiosos sugeriram algumas técnicas para resolver a questão. As melhores respostas obtidas foram compiladas e publicadas em (NEUMAIER, 2009). A função real do problema desafio é apresentada, a seguir:

$$f = \frac{a(w^2 + x^2 - y^2 - z^2) + 2b(xy - wz) + 2c(xz + wy)}{w^2 + x^2 + y^2 + z^2}, \quad (8\text{-}33)$$
$$a \in [7,9], b \in [-1,1], c \in [-1,1], w \in [-0,9, -0,6],$$
$$x \in [-0,1, 0,2], y \in [0,3, 0,7], z \in [-0,2, 0,1].$$

De acordo com NEUMAIER (2009) a função (8-33) foi mencionada por Nate Hayes e surgiu a partir de um problema na área de visão computacional. Para efeito de comparação e referência os

[11] Intervalo que envolve a imagem da função.

limites mínimo e máximo do envelope exato do problema desafio foram calculados por Mihaly Markot e Dan Zuras e são dados por:

$$\begin{aligned} \text{min\_ref} &= -2{,}9561, \\ \text{max\_ref} &= 8{,}0094. \end{aligned} \tag{8-34}$$

Com isso, a largura do envelope de referência é

$$\text{larg\_ref} = 10{,}9655. \tag{8-35}$$

Calculando-se a função de inclusão natural diretamente sobre f obtém-se como o envelope $[-7{,}4889,\ 19{,}2889]$, cuja largura é 244% da larg_ref. Dentre as soluções apresentadas em (NEUMAIER, 2009), a que obteve melhor resultado retornou o seguinte envelope $[-3{,}1073,\ 8{,}1089]$, cuja largura é 102,3% da larg_ref.

Com a aplicação do SNIF-MOGPA uma das aproximações obtidas para (8-33) é a expressão:

$$\begin{aligned} f_{aproximada} = {} & a(w^2 - 2y^2 + 1) - \frac{8y^2}{w^2} + 4cw + \frac{2x^2 - \frac{2y^2}{w^2} - 2y^2 + bx^2y + 4}{3y^2 + 1} \\ & + a(w^2 - y^2 - z^2 + 1) - \frac{c + \frac{4y^2}{w^2} - x^2 + y^2 + z^2 + cx^2yz - \frac{3283}{400}}{3y^2 + 1} \\ & - \frac{4y^2 + z^2}{w^2} + 2x^2 - 2y^2 - 6z^2 - \frac{c - x + x^3y^2 + \frac{y^2 + 2z^2}{w^2} + 2z^2 - byz^2 - 4}{2y^2 + 1} + 4. \end{aligned} \tag{8-36}$$

A inclinação e o intercepto de (8-36) são 0,1890 e -1,5239, respectivamente. O erro quadrático médio no treinamento foi de 0,0622 e na validação de 0,0642. A função de inclusão natural obtida a partir da $f_{aproximada}$ para o problema desafio gera o seguinte envelope $[-2{,}9567,\ 8{,}0094]$, cuja largura é 100,005% da larg_ref.

O SNIF-MOGPA foi configurado conforme os parâmetros da Tabela 8-20 para lidar com o problema desafio. A medida de *fitness* relacionada à complexidade da expressão é diferente do algoritmo original, pois foi utilizada a largura do intervalo de saída em relação a larg_ref. Além disso, à população inicial adicionou-se um indivíduo para representar a função de inclusão natural obtida a partir de f. Adicionalmente, utilizou-se a técnica de *output bounds* proposta em (PENNACHIN *et al.*, 2010) a fim de eliminar indivíduos (expressões) cuja largura do intervalo de saída excede o intervalo desejado $[\text{min\_ref} - \alpha r, \text{max\_ref} + \alpha r]$, sendo $r = (\text{max\_ref} - \text{min\_ref})/2$ e α um parâmetro. O parâmetro α não se manteve fixo, fez-se uma variação de forma a diminuir os limites do intervalo desejado ao longo das gerações. Até a centésima geração estabeleceu-se α=1, nas cem gerações seguintes configurou-se α=0,5 e no restante das gerações teve-se α=0,1.

Tabela 8-20 – Parâmetros do SNIF-MOGPA aplicado ao problema desafio.

Parâmetro	Valor
Número máximo de gerações	300
Tamanho da população	500
Taxa de cruzamento, mutação e reprodução	0,9, 0,1 e 0,2, respectivamente
Taxa de cruzamento entre nós internos	0,8
Conjunto de funções	+, -, *, / e ^2
Conjunto de terminais	a, b, c, w, x, y, z e constante efêmera
Intervalo das constantes efêmeras reais e inteiras	]-10, 10[e {1, 2, …, 5}
Tamanho do torneio	5
Altura máxima da árvore durante a execução	17
Altura máxima da árvore na geração inicial	4
Método de geração da população inicial	*Ramped half-and-half*
Critério de parada	Número máximo de gerações
Medida de *fitness*	Erro quadrático médio com *fitness scaling* e largura do intervalo de saída em relação a larg_ref
Conjunto de treinamento - *fitness cases*	4000 amostras
Conjunto de validação	500 amostras em um único grupo

A Tabela 8-21 apresenta o retorno de 10 execuções do SNIF-MOGPA. Para cada execução, os dados da expressão com menor largura em relação a larg_ref são mostrados. Verifica-se que todas as aproximações retornadas são válidas de acordo com o Wilcoxon *rank sum test* com nível de confiança de 95% (p-valor $\geq$ 0,05). Além disso, possuem coeficiente de determinação R2 bem próximo de 1 e largura do intervalo de saída bem próximas da largura de referência larg_ref.

Tabela 8-21 – Resultados de dez execuções do SNIF-MOGPA para o problema desafio.

Execução	Treinamento			Validação			Largura em relação a larg_ref (%)
	MSE	R2	p-valor	MSE	R2	p-valor	
1	0,0622	0,9805	0,7874	0,0642	0,9795	0,9840	100,005
2	0,0511	0,9842	0,7028	0,0551	0,9848	0,9254	100,005
3	0,0488	0,9849	0,6929	0,0461	0,9848	0,7850	100,015
4	0,0506	0,9843	0,7042	0,0472	0,9858	0,9171	100,012
5	0,0645	0,9793	0,7709	0,0686	0,9769	0,8266	100,001
6	0,0518	0,9839	0,6966	0,0527	0,9834	0,7854	100,075
7	0,0618	0,9808	0,6701	0,0555	0,9797	0,6858	100,065
8	0,0555	0,9828	0,6780	0,0563	0,9827	0,7671	100,094
9	0,0499	0,9845	0,6930	0,0519	0,9834	0,7805	100,171
10	0,0676	0,9790	0,7740	0,0705	0,9783	0,9943	100,024
Média	0,0564	0,9824	0,7170	0,0568	0,9819	0,8451	100,047
Desvio Padrão	0,0070	0,0023	0,0432	0,0084	0,0031	0,1035	0,055

Para efeitos comparativos, resolveu-se o problema desafio utilizando-se ANN-RBF via a função *newrb* disponível no ANN *Toolbox* do Matlab®. Para a obtenção das aproximações utilizou-se uma ANN-RBF com 100 e outra com 1000 neurônios na camada escondida. O restante dos parâmetros foi estabelecido pelos valores *default* da função *newrb*. A Tabela 8-22 mostra os resultados obtidos. Verifica-se que para ANN-RBF com 100 neurônios o MSE das aproximações apresentou a mesma ordem de magnitude que o MSE alcançado pelas aproximações com SNIF-MOGPA, contudo a ANN-RBF com 1000 neurônios encontrou aproximações mais acuradas. Porém, tanto para 100 quanto para 1000 neurônios, a largura do intervalo de saída das aproximações obtidas via ANN-RBF é

significativamente maior que as das expressões encontradas pelo SNIF-MOGPA, sendo inclusive, bem maior do que o envelope da própria função de inclusão natural diretamente sobre f. Isso torna inviável o uso das aproximações realizadas por ANN-RBF.

Tabela 8-22 – Resultados ANN-RBF para o problema desafio.

	Treinamento			Validação			Largura em relação a larg_ref (%)
	MSE	R2	p-valor	MSE	R2	p-valor	
ANN-RBF 100	0,0147	0,9954	0,9650	0,0166	0,9947	0,9851	3,81e+4
ANN-RBF 1000	3,6e-5	0,9999	0,9988	0,0001	0,9999	0,9936	1,26e+7

Nesta subseção, discutiu-se o uso do SNIF-MOGPA para encontrar uma função aproximada com o menor valor de MSE possível em relação à função real f e cujo intervalo de saída da função de inclusão natural fosse o mais próximo possível do intervalo do envelope de referência. Como resultado obtiveram-se aproximações para f com erro quadrático médio da ordem de 10^{-2}, cujo intervalo de saída da função de inclusão natural é mais estreito do que o fornecido por f.

8.4 Fronteira Ideal de Maximização

Para exemplificar $\mathbf{F}^{\max}$, essa foi utilizada no algoritmo determinístico robusto estritamente intervalar [I]RMOA II proposto em (SOARES, 2008). Executa-se o [I]RMOA II ora com $\mathbf{u}^{\max}$ ora com $\mathbf{F}^{\max}$ para fins de comparação.

Problema Teste I

O problema teste I refere-se ao exemplo 12 apresentado em (SOARES, 2008) e é dado como segue:

$$\min_{\mathbf{x}\in[-\mathbf{2},\mathbf{2}]} \max_{\|\mathbf{p}\|\leq 1} \quad \begin{matrix} f_1(\mathrm{x},\mathrm{p}_1) = -\mathrm{x}+\mathrm{p}_1, \\ f_2(\mathrm{x},\mathrm{p}_2) = \mathrm{x}+\mathrm{p}_2. \end{matrix} \tag{8-37}$$

Neste problema irrestrito, o resultado do [I]RMOA II com $\mathbf{u}^{\max}$ é mostrado na Figura 8-13. A fronteira robusta (asteriscos) está localizada fora da imagem viável (caixas em cinza claro e cinza escuro). A parte em cinza escuro da imagem viável refere-se às caixas que têm parte viável e parte inviável. As caixas em cinza claro são totalmente viáveis. Os seguintes parâmetros foram utilizados na execução do algoritmo: $\varepsilon_{\mathrm{p}} = 0{,}2$ e $\varepsilon_{\mathrm{x}} = 0{,}5$.

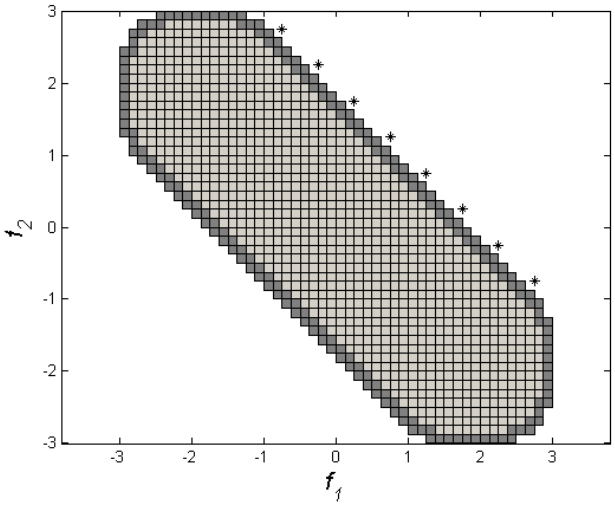


Figura 8-13 – Imagem viável e fronteira robusta do problema teste I utilizando $\mathbf{u}^{\max}$.

Aplicando-se o [I]RMOA II com $\mathbf{F}^{\max}$ e utilizando-se dos mesmos parâmetros descritos anteriormente, tem-se que a fronteira robusta (asteriscos) está contida na imagem viável. Esse resultado pode ser visto na Figura 8-14. Percebe-se que nesse caso a fronteira robusta é composta na verdade por várias (oito neste exemplo) fronteiras não-dominadas entre si. Possibilidade que foi discutida na Seção 7.2 logo após a definição de $\mathbf{F}^{\max}$.

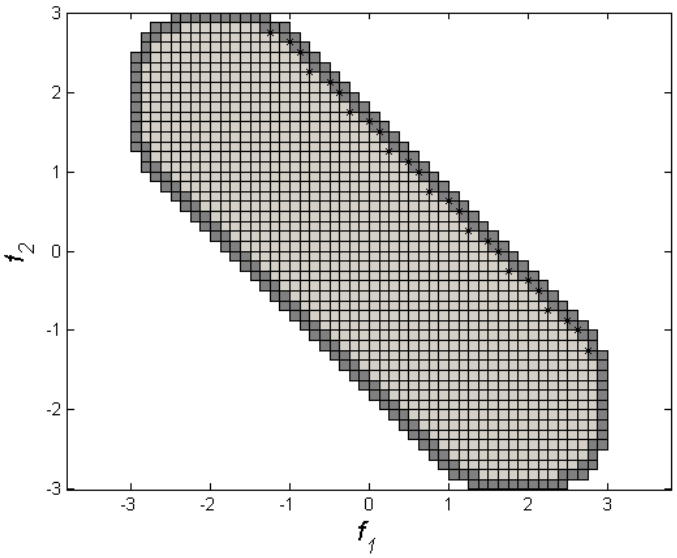


Figura 8-14 – Fronteira robusta do problema teste I utilizando $\mathbf{F}^{\max}$.

A Figura 8-15 exibe os resultados da execução do IRMOA-II com os parâmetros $\varepsilon_p = 0{,}2$ e $\varepsilon_x = 0{,}1$. Devido à precisão mais acurada, a quantidade de soluções retornadas é maior (sessenta e quatro). Os círculos em branco indicam a fronteira robusta obtida quando se utiliza $\mathbf{u}^{\max}$. Os asteriscos pretos indicam os conjuntos de pontos não-dominados que formam a fronteira robusta quando se utiliza $\mathbf{F}^{\max}$. Existem 64 conjuntos de pontos não-dominados que agrupados formam a fronteira robusta.

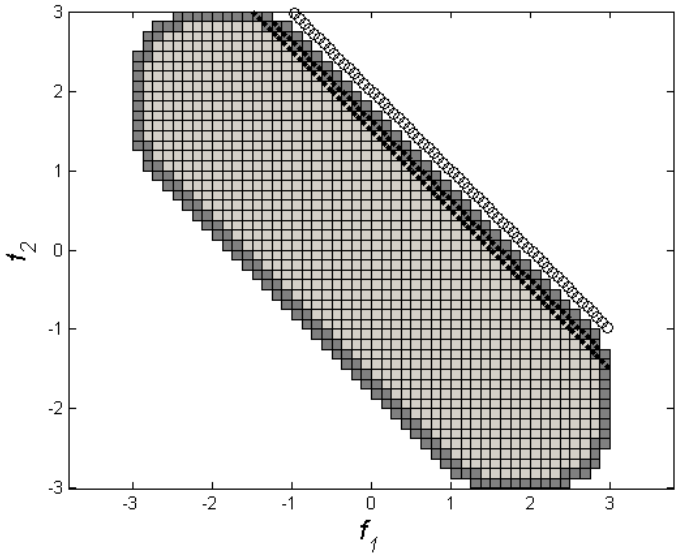


Figura 8-15 – Fronteira robusta do problema teste I utilizando $\mathbf{u}^{\max}$ (círculos brancos) e $\mathbf{F}^{\max}$ (asteriscos) com $\varepsilon_x = 0{,}1$.

Problema Teste II

O problema teste II, denominado SCH2, foi extraído do trabalho de (SOARES, 2008) e consiste na incorporação de incertezas na função Schaffer 2, sendo definido por:

$$\min_{\mathbf{x}\in[-5,10]} \max_{\mathbf{p}\in[-0{,}1,\ 0{,}1]} \quad f_1(\mathrm{x},\mathbf{p}) = \begin{cases} -(\mathrm{x}+\mathrm{p}_1), & \text{se } \mathrm{x} \le 1 \\ (\mathrm{x}+\mathrm{p}_1)-2, & \text{se } 1 < \mathrm{x} \le 3 \\ 4-(\mathrm{x}+\mathrm{p}_1), & \text{se } 3 < \mathrm{x} \le 4 \\ (\mathrm{x}+\mathrm{p}_1)-4, & \text{se } \mathrm{x} > 4; \end{cases} \tag{8-38}$$

$$f_2(\mathrm{x},\mathbf{p}) = \left((\mathrm{x}+\mathrm{p}_1)-5\right)^2 + \mathrm{p}_2.$$

O resultado encontrado pelo algoritmo [I]RMOA II após ser aplicado na resolução do problema (8-38) com $\varepsilon_p = 0{,}2$ e $\varepsilon_x = 0{,}1$ é mostrado na Figura 8-16. Os pontos contínuos em cinza claro indicam a imagem robusta viável. Os asteriscos em cinza escuro representam a fronteira robusta obtida quando se usa $\mathbf{u}^{\max}$ e os asteriscos pretos quando utiliza-se $\mathbf{F}^{\max}$. Embora o número de asteriscos pretos seja maior (isso ocorre uma vez que cada solução associada a $\mathbf{F}^{\max}$ é um conjunto de pontos não-dominados), ressalta-se que a fronteira em cinza escuro possui uma solução a mais do que a fronteira preta. Verifica-se que a fronteira gerada com a utilização de $\mathbf{F}^{\max}$ domina a maior parte da fronteira obtida com o uso de $\mathbf{u}^{\max}$ e, dessa forma, é mais próxima da imagem robusta viável. Sendo assim, pode-se concluir para este problema, que a maximização das incertezas utilizando-se $\mathbf{F}^{\max}$ é menos conservadora que pelo uso de $\mathbf{u}^{\max}$. Na Figura 8-16 à direita tem-se uma ampliação de parte das fronteiras, note que cada asterisco cinza escuro (exceto o mais à esquerda) é relacionado a dois asteriscos pretos. Cada asterisco cinza escuro é formado pelo pior valor de cada dupla de asteriscos pretos associados a esse.

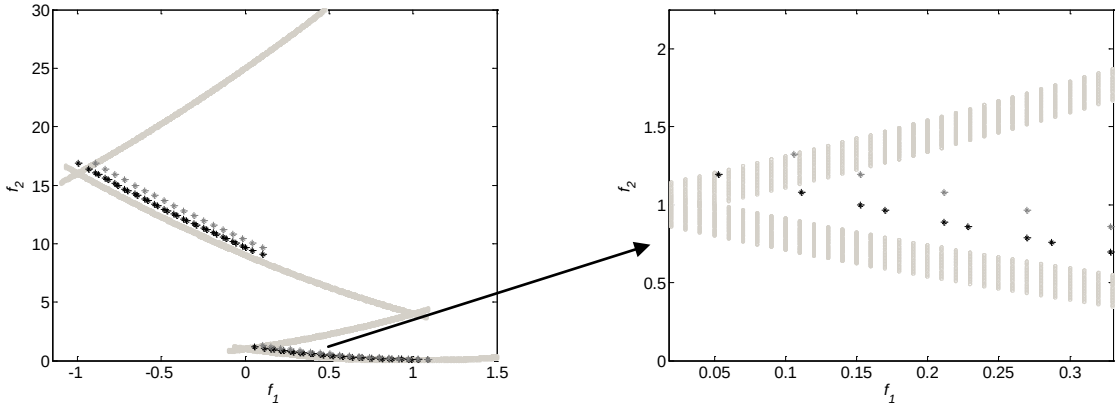


Figura 8-16 – Fronteira robusta do problema SCH2 utilizando $\mathbf{u}^{\max}$ (asteriscos em cinza escuro) e $\mathbf{F}^{\max}$ (asteriscos pretos) com $\varepsilon_x = 0{,}1$.

Problema Teste III

O problema teste III, denominado FON, foi extraído do trabalho de (SOARES, 2008) e é formulado como segue:

$$\min_{\mathbf{x}\in[-4,4]} \max_{\mathrm{p}\in[1{,}1,\ 1{,}3]} \quad f_1(\mathbf{x},\mathrm{p}) = 1 - e^{-\sum_{i=1}^{n}\left(x_i - \frac{p}{\sqrt{n}}\right)^2} \tag{8-39}$$

$$f_2(\mathbf{x}, \mathrm{p}) = 1 - e^{-\sum_{i=1}^{n}\left(x_i + \frac{p}{\sqrt{n}}\right)^2}.$$

Considerando-se $n = 2$, o resultado encontrado pelo algoritmo [I]RMOA II após ser aplicado na resolução do problema (8-39) com $\varepsilon_{\mathrm{p}} = 0{,}025$ e $\varepsilon_{\mathrm{x}} = 0{,}1$ é mostrado na Figura 8-17. Os pontos em cinza claro indicam a imagem robusta viável. Os círculos com borda preta representam a fronteira robusta obtida quando se usa $\mathbf{u}^{\max}$ e os pontos em azul quando utiliza-se $\mathbf{F}^{\max}$. Nesse exemplo, as duas fronteiras obtidas são bem semelhantes, exceto nas extremidades, nas quais a fronteira com $\mathbf{F}^{\max}$ apresenta ligeira vantagem.

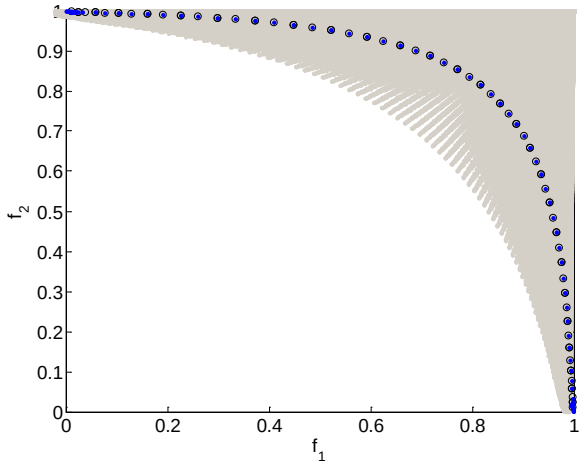


Figura 8-17 – Fronteira robusta do problema FON utilizando $\mathbf{u}^{\max}$ (círculos) e $\mathbf{F}^{\max}$ (pontos em azul) com $\varepsilon_{\mathrm{x}} = 0{,}1$.

Problema Teste IV

O problema teste IV, denominado ZDT2, foi extraído do trabalho de (SOARES, 2008) e é formulado como segue:

$$\begin{aligned}
&\min_{\mathbf{x}\in[0,1]} \quad \max_{\mathrm{p}\in[-0{,}05,\ 0{,}05]} && f_1(\mathbf{x}, \mathrm{p}) = \mathrm{x}_1 \\
&&& f_2(\mathbf{x}, \mathrm{p}) = h'(\mathbf{x}, p) \times h(f_1, h', p) \\
&s.a && g(\mathbf{x}) = \sqrt{(\mathrm{x}_1 - 7)^2 + {\mathrm{x}_2}^2 + \cdots + {\mathrm{x}_\mathrm{n}}^2} > 0.1 \\
&sendo && h'(\mathbf{x}, p) = 1 + \frac{9}{n-1}\left(\sum_{i=2}^{n} \mathrm{x}_i\right) + p \\
&&& h(f_1, h', p) = 1 - (f_1/h')^2 + p.
\end{aligned} \tag{8-40}$$

Considerando-se $\mathrm{n} = 2$, o resultado encontrado pelo algoritmo [I]RMOA II após ser aplicado na resolução do problema (8-40) com $\varepsilon_{\mathrm{p}} = 0{,}02$ e $\varepsilon_{\mathrm{x}} = 0{,}1$ é mostrado na Figura 8-16. Os pontos em cinza claro indicam a imagem robusta viável. Os círculos com borda preta representam a fronteira robusta obtida quando se usa $\mathbf{u}^{\max}$ e os pontos em cinza escuro quando utiliza-se $\mathbf{F}^{\max}$. Nesse exemplo, as duas fronteiras obtidas levam aos mesmos pontos.

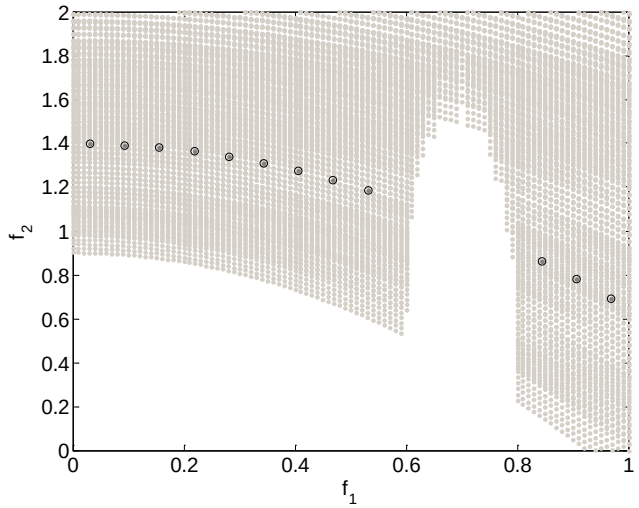


Figura 8-18 – Fronteira robusta do problema ZDT2 utilizando $\mathbf{u}^{\max}$ (círculos) e $\mathbf{F}^{\max}$ (pontos em cinza escuro) com $\varepsilon_x = 0{,}1$.

8.5 Aplicação do IRMOEA-M

Nesta seção, apresenta-se o resultado da aplicação do IRMOEA-M aos problemas teste SCH2, FON, ZDT2 e a um problema de engenharia, denominado problema da viga. Para fins comparativos utilizou-se o IRMOEA-M tanto com $\mathbf{u}^{\max}$ quanto com $\mathbf{F}^{\max}$. Para cada problema são realizadas 21 execuções do IRMOEA-M com $\mathbf{u}^{\max}$ e 21 execuções do IRMOEA-M com $\mathbf{F}^{\max}$. A configuração do IRMOEA-M em cada execução é a seguinte: 50 gerações, 50 indivíduos, seleção por torneio de tamanho 2, probabilidade de cruzamento igual a 0,9 e probabilidade de mutação igual a 0,1. Os valores usados para o parâmetro ε_p são mostrados na Tabela 8-23. Os resultados obtidos são comparados por meio da noção de *Attainment Surface* (ASU). Conforme LÓPEZ-IBÁNEZ *et al.* (2010) o resultado de um algoritmo multiobjetivo não determinístico para um problema particular pode ser representado por uma *Attainment Function* (AF). A AF relativa a um algoritmo, em geral, é desconhecida. Entretanto, é possível derivar a *Empirical Attainment Function* (EAF) a partir dos resultados de diversas execuções independentes de um algoritmo. Nesta tese, utilizou-se a EAF de primeira ordem e, a partir dessa, obteve-se cada ASU.

Tabela 8-23 – Valor de ε_p para cada problema.

Problema	**Valor de ε_p**
SCH2	0,2
FON	0,03
ZDT2	0,02
Viga	0,6

Problema SCH2

A Figura 8-19 exibe algumas *Attainment Surfaces* (ASUs) obtidas como resultado para o problema SCH2. A Worst ASU é interpretada como o limite alcançado em todas as 21 execuções, isto é, a região dominada ou igual a essa Worst ASU sempre foi atingida pelo algoritmo nas 21 execuções. A Best ASU é interpretada como o limite alcançado em pelo menos uma das execuções. A região

compreendida entre a Best ASU e a Worst ASU serve como indicação da variabilidade dos resultados do algoritmo. Para o problema SCH2 verifica-se que em boa parte do espaço dos objetivos a Worst ASU do IRMOEA-M com $\mathbf{F}^{max}$ é melhor que a Best ASU do IRMOEA-M com $\mathbf{u}^{Max}$.

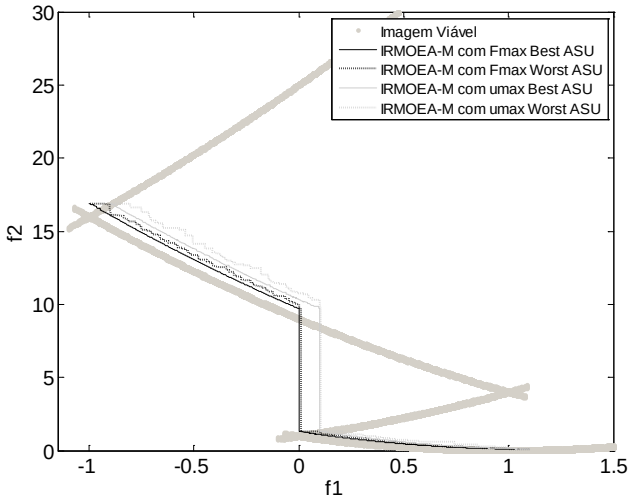


Figura 8-19 – ASUs para o problema SCH2 na formulação robusta *minimax*.

Problema FON

A Figura 8-20 exibe a Best e a Worst ASUs obtidas para o problema FON. Verifica-se que a variabilidade dos resultados para o IRMOEA-M com $\mathbf{F}^{max}$ é maior. Apesar disso, a Best ASU de ambos os algoritmos é bem próxima, com uma ligeira vantagem para o IRMOEA-M com $\mathbf{u}^{max}$ na região central das Best ASUs.

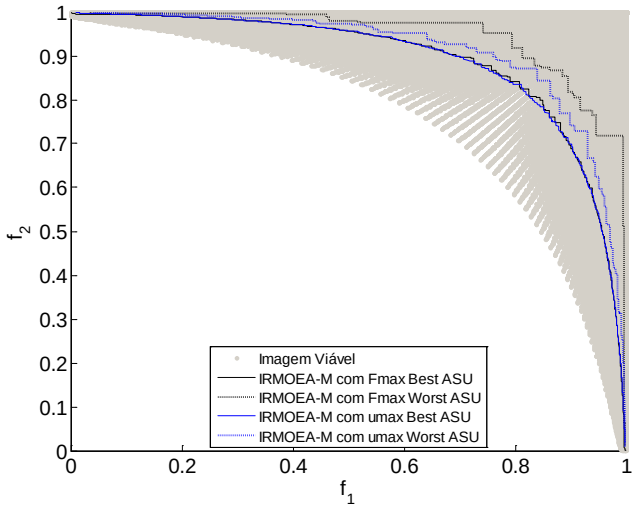


Figura 8-20 – ASUs para o problema FON na formulação robusta *minimax*.

A Figura 8-21 exibe a Mediana ASU do IRMOEA-M com $\mathbf{F}^{max}$ e do IRMOEA-M com $\mathbf{u}^{max}$ para o problema FON. A Mediana ASU é interpretada como o limite alcançado por metade das execuções. A localização da Mediana ASU fornece uma ideia da provável localização da saída em uma única

execução do algoritmo. Dessa forma, é na região central da Mediana ASU que o IRMOEA-M com $\mathbf{u}^{max}$ tem vantagem em relação ao IRMOEA-M com $\mathbf{F}^{max}$.

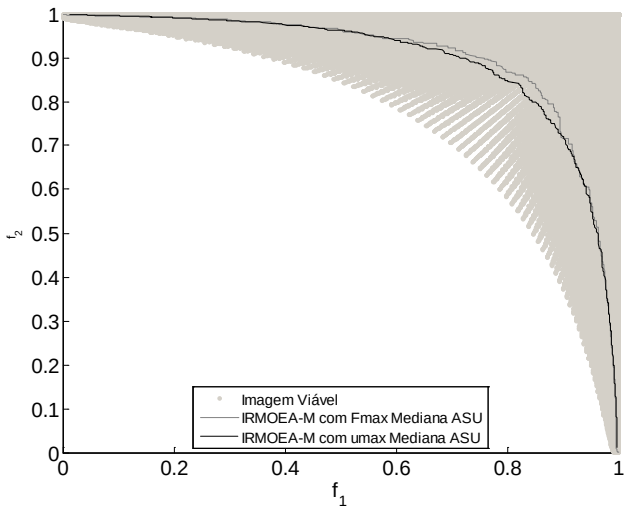


Figura 8-21 – Mediana ASUs para o problema FON na formulação robusta *minimax*.

Problema ZDT2

A Figura 8-22 exibe a Best e a Worst ASUs obtidas para o problema ZDT2. Verifica-se que a variabilidade dos resultados para o IRMOEA-M com $\mathbf{F}^{max}$ é maior em parte da saída. Apesar disso, a Best ASU de ambos os algoritmos é bem semelhante, exceto na região próxima à descontinuidade, na qual o IRMOEA-M com $\mathbf{u}^{max}$ apresenta ligeira vantagem.

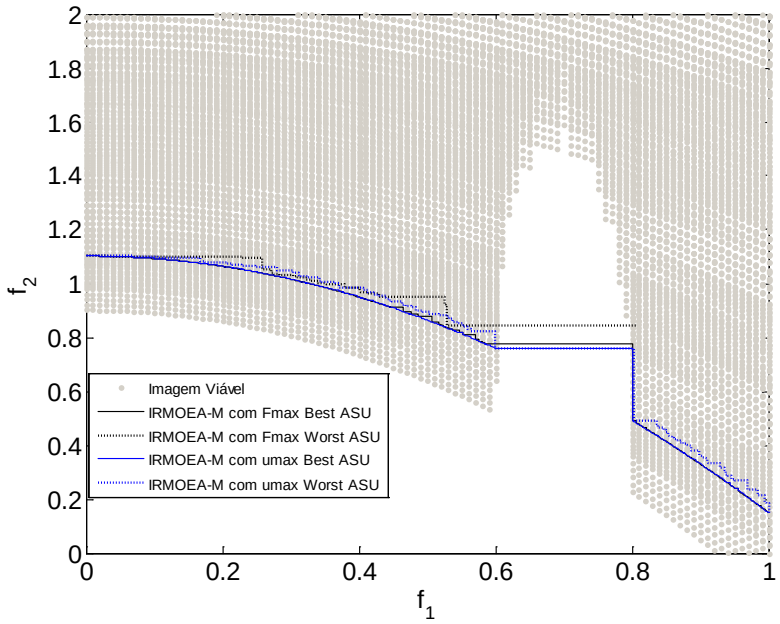


Figura 8-22 – ASUs para o problema ZDT2 na formulação robusta *minimax*.

A Figura 8-23 exibe a Mediana ASU do IRMOEA-M com $\mathbf{F}^{\max}$ e do IRMOEA-M com $\mathbf{u}^{\max}$ para o problema ZDT2. Nota-se que na região próxima à descontinuidade o IRMOEA-M com $\mathbf{u}^{\max}$ tem vantagem em relação ao IRMOEA-M com $\mathbf{F}^{\max}$.

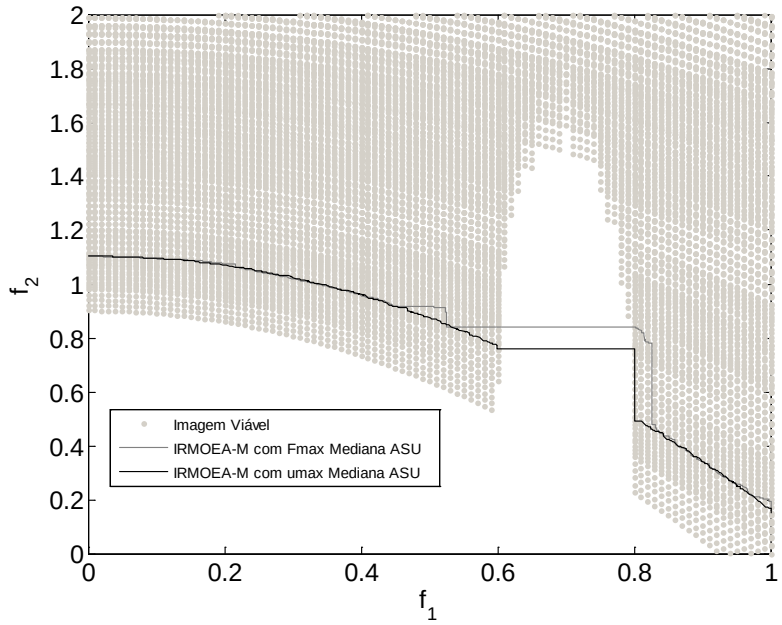


Figura 8-23 – Mediana ASUs para o problema ZDT2 na formulação robusta *minimax*.

Problema da Viga

O problema da viga ilustrado pela Figura 8-24 e extraído de (COELLO e CHRISTIANSEN, 1998) consiste em encontrar as dimensões (x_1, x_2, x_3 e x_4) para uma viga em seção I de modo a minimizar a área da seção transversal e o deslocamento estático máximo respeitando-se a restrição de projeto relacionada às cargas P e Q.

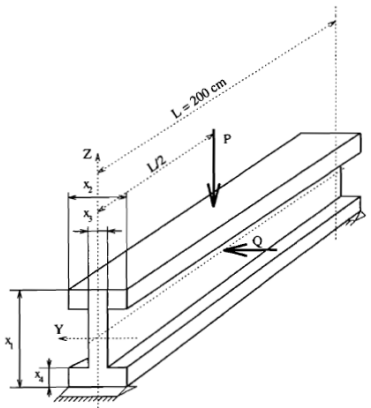


Figura 8-24 – Viga em seção I (COELLO e CHRISTIANSEN, 1998).

A versão robusta *minimax* do problema da viga é formulada como segue:

$$\begin{array}{c} \min \\ x_1 \in [10,80] \\ x_2 \in [10,50] \\ x_3 \text{ e } x_4 \in [0.9,5] \\ s.a \end{array} \quad \begin{array}{c} \max \\ p \in [-1,1] \end{array} \quad \begin{array}{c} f_1(\mathbf{x}, p), f_2(\mathbf{x}, p) \\ \\ g(\mathbf{x}, p) \end{array} \tag{8-41}$$

com

$$f_1(\mathbf{x}, \mathrm{p}) = 2(\mathrm{x}_2 + p)(\mathrm{x}_4 + p) + (\mathrm{x}_3 + p)\left((\mathrm{x}_1 + p) - 2(\mathrm{x}_4 + p)\right) \tag{8-42}$$

$$f_2(\mathbf{x}, \mathrm{p}) = \frac{60000}{(\mathrm{x}_3 + p)\left((\mathrm{x}_1 + p) - (\mathrm{x}_4 + p)\right)^3 + 2(\mathrm{x}_2 + p)(\mathrm{x}_4 + p)\left[4(\mathrm{x}_4 + p)^2 + 3(\mathrm{x}_1 + p)\left((\mathrm{x}_1 + p) - 2(\mathrm{x}_4 + p)\right)\right]} \tag{8-43}$$

$$\begin{aligned} \mathrm{g}(\mathbf{x}, \mathrm{p}) = &- \frac{180000(\mathrm{x}_1 + p)}{(\mathrm{x}_3 + p)\left((\mathrm{x}_1 + p) - (\mathrm{x}_4 + p)\right)^3 + 2(\mathrm{x}_2 + p)(\mathrm{x}_4 + p)\left[4(\mathrm{x}_4 + p)^2 + 3(\mathrm{x}_1 + p)\left((\mathrm{x}_1 + p) - 2(\mathrm{x}_4 + p)\right)\right]} \\ &+ 16 - \frac{15000(\mathrm{x}_2 + p)}{\left((\mathrm{x}_1 + p) - 2(\mathrm{x}_4 + p)\right)(\mathrm{x}_3 + p)^3 + 2(\mathrm{x}_4 + p)(\mathrm{x}_2 + p)^3} \geq 0 \end{aligned} \tag{8-44}$$

A Figura 8-25 exibe a Best e a Worst ASUs obtidas como resultado da aplicação do IRMOEA-M ao problema da viga. Verifica-se que em todo o espaço dos objetivos a Worst ASU do IRMOEA-M com $\mathbf{F}^{\max}$ é melhor que a Best ASU do IRMOEA-M com $\mathbf{u}^{\max}$.

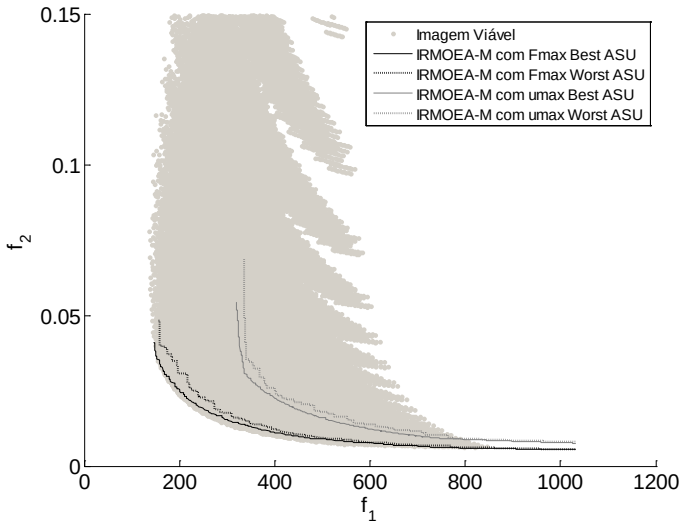


Figura 8-25 – ASUs para o problema da viga na formulação robusta *minimax*.

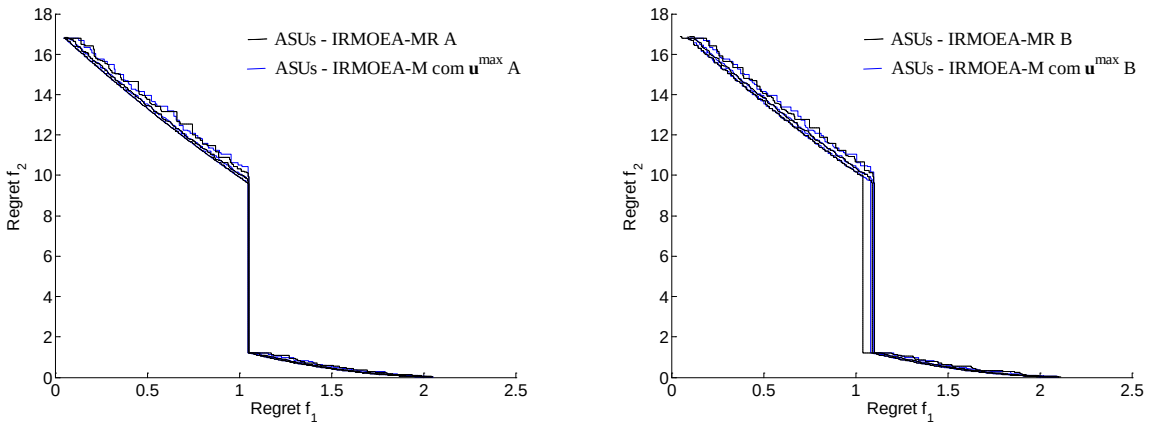
8.6 Aplicação do IRMOEA-MR

Nesta seção, apresenta-se o resultado da aplicação do IRMOEA-MR aos problemas teste SCH2, FON, ZDT2 e ao problema da viga, desta vez, formulados de acordo com o *minimax regret* conforme apresentado na Subseção 2.1.2. Utilizou-se o IRMOEA-MR em duas versões: A) com a etapa 2 (que é encontrar a melhor solução para cada caixa do subpavimento de **P**) sendo realizada por um algoritmo multiobjetivo robusto (IRMOEA-M), e B) com a etapa 2 sendo realizada por um algoritmo robusto mono-objetivo (IREA-M). Para cada problema são realizadas 21 execuções do IRMOEA-MR A e 21 execuções do IRMOEA-MR B. A configuração do IRMOEA-MR A e B em cada execução é a seguinte: 50 gerações, 50 indivíduos, seleção por torneio de tamanho 2, probabilidade de cruzamento igual a 0,9 e probabilidade de mutação igual a 0,1. Os valores usados para o parâmetro ε_p permanecem os mesmos da Tabela 8-23. A configuração do algoritmo evolucionário da etapa 2 (IRMOEA-M ou IREA-M) também é: 50 gerações, 50 indivíduos, seleção por torneio de tamanho 2,

probabilidade de cruzamento igual a 0,9 e probabilidade de mutação igual a 0,1. Para fins de comparação, calculou-se o *regret* das soluções robustas obtidas pelo IRMOEA-M com $\mathbf{u}^{max}$ na seção anterior, a fim de confrontar esse resultado com aqueles obtidos pelo IRMOEA-MR A e IRMOEA-MR B. Ressalta-se que no cômputo do *regret* das soluções obtidas pelo IRMOEA-M com $\mathbf{u}^{max}$ é necessário levar em consideração a melhor solução para cada caixa do subpavimento de **P**. Portanto tem-se o IRMOEA-M com $\mathbf{u}^{max}$ A e IRMOEA-M com $\mathbf{u}^{max}$ B, que consideram a melhor solução obtida pelo algoritmo multiobjetivo robusto e pelo algoritmo mono-objetivo robusto, respectivamente. Os resultados obtidos são comparados avaliando-se as ASUs do IRMOEA-MR A em relação as ASUs do IRMOEA-M com $\mathbf{u}^{max}$ A e as ASUs do IRMOEA-MR B em relação as ASUs do IRMOEA-M com $\mathbf{u}^{max}$ B para cada um dos problemas.

Problema SCH2

A Figura 8-26 exibe algumas ASUs obtidas como resultado para o problema SCH2 na formulação robusta *minimax regret*. Uma vez que o *regret* resulta em um número não negativo para cada um dos objetivos, ressalta-se que as ASUs obtidas estarão no quadrante em que todas as funções objetivo são não negativas. Na Figura 8-26 (a) tem-se, em pontilhado preto, as ASUs (Best, Mediana e Worst) para o IRMOEA-MR A e, em azul, as ASUs (Best, Mediana e Worst) para o IRMOEA-M com $\mathbf{u}^{max}$ A. Na Figura 8-26 (b) tem-se, em pontilhado preto, as ASUs (Best, Mediana e Worst) para o IRMOEA-MR B e, em azul, as ASUs (Best, Mediana e Worst) para o IRMOEA-M com $\mathbf{u}^{max}$ B. Verifica-se, tanto na Figura 8-26 (a) quanto na Figura 8-26 (b), que as respectivas ASUs têm desempenho parecido.



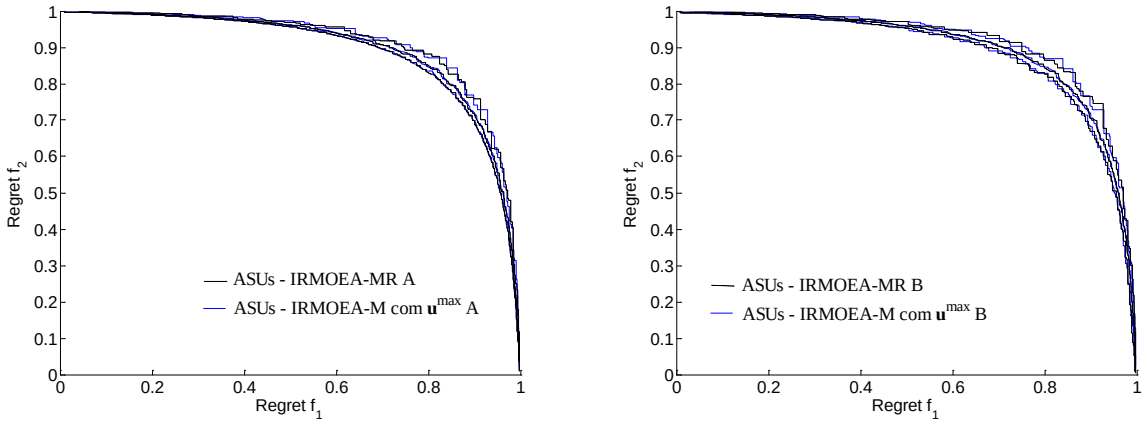
(a) (b)

Figura 8-26 – ASUs para o problema SCH2 na formulação robusta *minimax regret*.

Problema FON

Na Figura 8-27 (a) tem-se, em pontilhado preto, as ASUs (Best, Mediana e Worst) para o IRMOEA-MR A e, em azul, as ASUs (Best, Mediana e Worst) para o IRMOEA-M com $\mathbf{u}^{max}$ A. Na Figura 8-27 (b) tem-se, em pontilhado preto, as ASUs (Best, Mediana e Worst) para o IRMOEA-MR B e, em azul,

as ASUs (Best, Mediana e Worst) para o IRMOEA-M com $\mathbf{u}^{max}$ B. Nota-se, tanto na Figura 8-27 (a) quanto na Figura 8-27 (b), que as respectivas ASUs possuem desempenho semelhante.

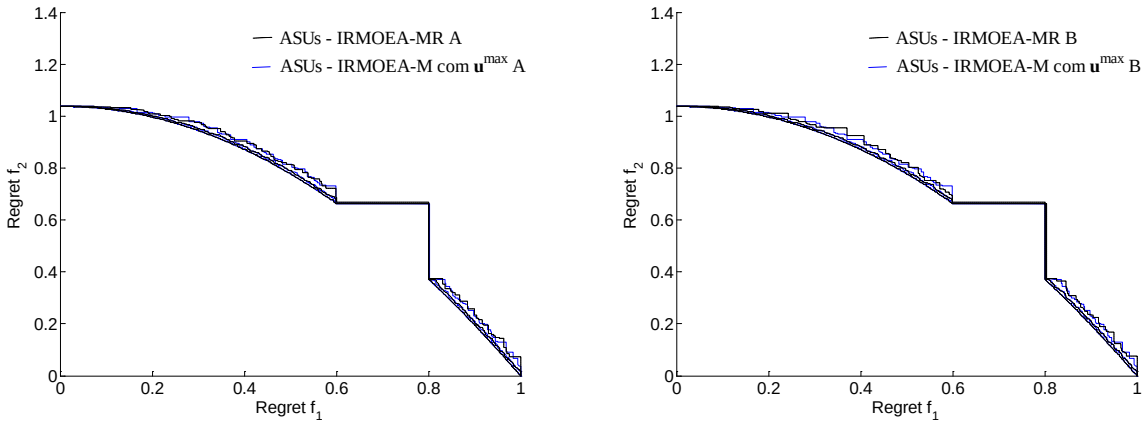


(a) (b)

Figura 8-27 – ASUs para o problema FON na formulação robusta *minimax regret*.

Problema ZDT2

Na Figura 8-28 (a) tem-se, em pontilhado preto, as ASUs (Best, Mediana e Worst) para o IRMOEA-MR A e, em azul, as ASUs (Best, Mediana e Worst) para o IRMOEA-M com $\mathbf{u}^{max}$ A. Na Figura 8-28 (b) tem-se, em pontilhado preto, as ASUs (Best, Mediana e Worst) para o IRMOEA-MR B e, em azul, as ASUs (Best, Mediana e Worst) para o IRMOEA-M com $\mathbf{u}^{max}$ B. Verifica-se, tanto na Figura 8-28 (a) quanto na Figura 8-28 (b), que as respectivas ASUs não apresentam diferenças significativas.

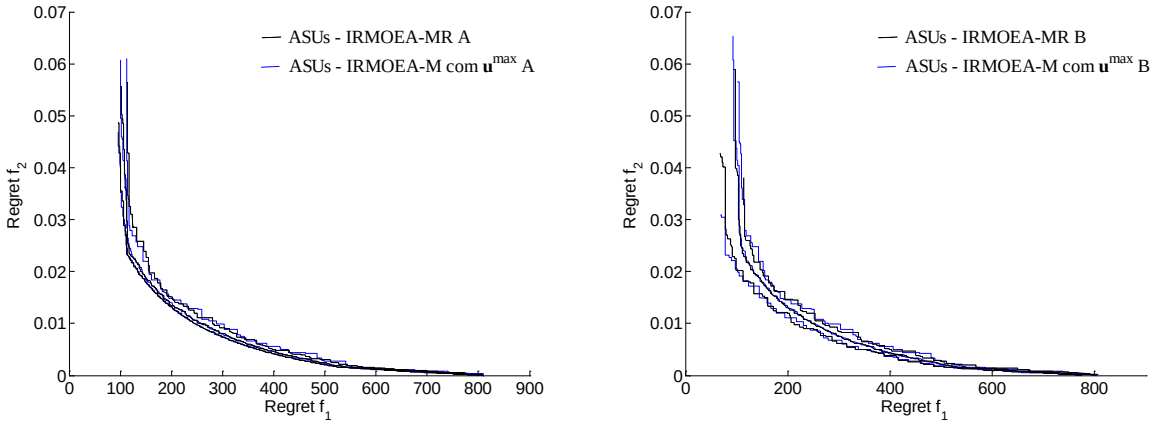


(a) (b)

Figura 8-28 – ASUs para o problema ZDT2 na formulação robusta *minimax regret*.

Problema da Viga

Na Figura 8-29 (a) tem-se, em pontilhado preto, as ASUs (Best, Mediana e Worst) para o IRMOEA-MR A e, em azul, as ASUs (Best, Mediana e Worst) para o IRMOEA-M com $\mathbf{u}^{max}$ A. Na Figura 8-29 (b) tem-se, em pontilhado preto, as ASUs (Best, Mediana e Worst) para o IRMOEA-MR B e, em azul, as ASUs (Best, Mediana e Worst) para o IRMOEA-M com $\mathbf{u}^{max}$ B. Verifica-se, tanto na Figura 8-29 (a) quanto na Figura 8-29 (b), que as respectivas ASUs possuem desempenho semelhante.



(a) (b)

Figura 8-29 – ASUs para o problema da viga na formulação robusta *minimax regret*.

A comparação das ASUs nos problemas SCH2, FON, ZDT2 e viga tanto para o IRMOEA-MR A quanto para o IRMOEA-MR B em relação ao IRMOEA-M com $\mathbf{u}^{\max}$ A e IRMOEA-M com $\mathbf{u}^{\max}$ B, respectivamente, mostrou que não ocorre diferença significativa nas respectivas ASUs. Dessa forma, o IRMOEA-MR é uma alternativa válida para a resolução de problemas robustos multiobjetivo quando esses são modelados no formato *minimax regret* estabelecido na Subseção 2.1.2. Nessa alternativa, ao fim do processo de otimização, obtêm-se pontos não-dominados no espaço dos objetivos que se referem ao melhor máximo *regret* (considerando-se todos os subpavimentos de incerteza) de cada solução robusta em cada uma das dimensões do problema. O valor de cada ponto não-dominado indica o quão distante (no pior caso) essa solução está em relação ao melhor possível. Essa informação é relevante em problemas robustos uma vez que fornece uma expectativa do quanto se pode melhorar em cada uma das dimensões. Assim, se possível, pode-se tentar amenizar as incertezas que afetam a dimensão cujo valor de máximo *regret* é mais alto a fim de se obter melhores soluções.

8.7 Conclusão

O enfoque deste capítulo foi na aplicação dos algoritmos, conceitos e definições propostas no capítulo anterior. Inicialmente, mostrou-se que os algoritmos SNIF-GPA e SNIF-MOGPA, fundamentados em PG, são capazes de encontrar boas funções de inclusão para problemas nas áreas de engenharia de controle e eletromagnetismo. Em seguida, a fronteira ideal de maximização foi utilizada com sucesso no [I]RMOA II para encontrar soluções robustas, mostrando-se vantajosa em relação ao ponto ideal de maximização. Adicionalmente, realizaram-se experimentos considerando-se diversos problemas para validar o IRMOEA-M e o IRMOEA-MR. Ambos mostraram-se capazes de retornar soluções robustas adequadas conforme as formulações das subseções 2.1.1 e 2.1.2, respectivamente. Ressalta-se que, além dos resultados apresentados ao longo deste capítulo, as ideias apresentadas nesta tese, principalmente, o que foi detalhado na Seção 7.2, também contribuíram para o desenvolvimento do algoritmo IMOEA-DFR (*Interval Multi-objective Evolutionary Algorithm for Distribution Feeder*

Reconfiguration), mais detalhes em (BARBOSA, 2012). A aplicação do IMOEA-DFR na reconfiguração robusta de sistemas de distribuição resultou no artigo intitulado "*Robust Feeder Reconfiguration in Radial Distribution Networks*" submetido a um periódico internacional.

9. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

9.1 Conclusões

Atualmente, verifica-se que o escopo do processo de otimização multiobjetivo torna-se mais amplo e requer metodologias capazes de obter soluções que funcionem perfeitamente em ambientes incertos. Com isso, nesta tese, investigaram-se formas eficientes de utilizar EAs e IA para encontrar soluções robustas no cenário de pior caso das incertezas para problemas multiobjetivo tanto pelo *minimax* quanto pelo *minimax regret*. Inicialmente, consultou-se a literatura a fim de identificar as principais dificuldades encontradas pelas metodologias existentes ao lidarem com o RMOPs. Em seguida, foram propostos conceitos, definições e formulações de modo a construir novos algoritmos evolucionários eficientes para otimização robusta multiobjetivo. Dentre as contribuições apresentadas e discutidas estão os algoritmos SNIF-GPA e SNIF-MOGPA para encontrar funções de inclusão aproximadas, um novo modo de computar a maximização das incertezas que é a fronteira ideal de maximização, a definição do operador de *regret* para problemas multiobjetivo e os algoritmos IRMOEA-M e IMOEA-MR para solucionar RMOPs.

Os algoritmos SNIF-GPA e SNIF-MOGPA permitem lidar com a dificuldade de encontrar funções de inclusões para as funções objetivo e restrições do RMOP. Ambos utilizam a técnica evolucionária de programação genética a fim de obter expressões analíticas para as funções de otimização, atendendo exigências do Teorema 2 e, dessa forma, permitem a obtenção de boas funções de inclusão para serem usadas nos algoritmos intervalares robustos. Nas seções 8.1 e 8.2 foram realizados experimentos em problemas nas áreas de engenharia de controle e eletromagnetismo, respectivamente. Os resultados mostraram que o SNIF-GPA e o SNIF-MOGPA foram capazes de obter boas funções de inclusões, as quais auxiliaram assertivamente os algoritmos de otimização no processo de busca por soluções robustas. Inclusive, no caso do problema TEAM 22 robusto houve significativa economia de tempo, de simulações computacionais e, ainda sim, a qualidade da fronteira robusta gerada foi preservada. Na Seção 8.3, apresentou-se uma variante do SNIF-MOGPA, que gerou resultados expressivos quando os valores mínimo e máximo da inclusão de referência são conhecidos. Enfim, o uso das funções de inclusão aproximadas mostrou-se vantajoso, principalmente, nas situações em que os valores das funções de otimização são obtidos por meio de custosas simulações computacionais.

Na Seção 8.4, para avaliar o conceito proposto de fronteira ideal de maximização, utilizou-se $\mathbf{F}^{\max}$ em um algoritmo robusto estritamente intervalar, denominado [I]RMOA II. Os resultados mostraram que, em todos os problemas testados, o uso de $\mathbf{F}^{\max}$ conduziu o [I]RMOA II a um desempenho igual ou superior quando comparado ao desempenho do [I]RMOA II com o ponto ideal de maximização $\mathbf{u}^{\max}$. Na Seção 8.5, comparou-se o IRMOEA-M com $\mathbf{F}^{\max}$ frente ao IRMOEA-M com $\mathbf{u}^{\max}$. Os resultados mostraram que, em alguns problemas, o uso de $\mathbf{F}^{\max}$ mostrou-se bem vantajoso. Porém,

em outros problemas teste, os resultados foram similares, com alguma vantagem para $\mathbf{u}^{\max}$ em algumas regiões da melhor fronteira robusta obtida.

Os algoritmos IRMOEA-M e IRMOEA-MR foram avaliados nas seções 8.5 e 8.6, respectivamente. Os experimentos mostraram que, nos problemas teste e no problema da viga, ambos os algoritmos são capazes de obter soluções robustas adequadas conforme as formulações apresentadas nas subseções 2.1.1 e 2.1.2, respectivamente. Salienta-se que os problemas abrangeram situações diversas como: incertezas paramétricas, ruído na função objetivo, restrições, fronteira descontínua, convexa e côncava. Ressalta-se que não se encontrou na literatura nenhum algoritmo com as características do IRMOEA-MR: evolucionário, tratamento intervalar das incertezas, dedicado à otimização robusta multiobjetivo e com noção de robustez dada pelo *minimax regret*.

9.2 Trabalhos Futuros

Os principais pontos que podem ser explorados visando o avanço dos assuntos investigados nesta tese são os seguintes:

- Incorporação de mecanismos ao SNIF-GPA e SNIF-MOGPA para que esses sejam capazes de detectar se é mais vantajoso (em termos de qualidade e/ou complexidade da aproximação) particionar o conjunto de treinamento e aproximar cada partição, em separado, por uma expressão analítica. O desafio principal é determinar os pontos ideais do domínio (variáveis de entrada) para que seja(m) realizado(s) o(s) particionamento(s);
- Estratégia para gerar as amostras de dados para o SNIF-GPA e SNIF-MOGPA. Se é conhecido de antemão que o modelo aproximado será usado para otimização e tem-se a formulação matemática do problema, pode-se, de acordo com o valor de retorno da função, privilegiar que regiões promissoras (maior ou menor valor de retorno, dependendo da formulação matemática do problema) tenham maior densidade de amostras. Afinal, é relevante que o modelo aproximado seja mais acurado na região promissora;
- Realização do tratamento das incertezas por meio da *Modal Interval Analysis*, uma extensão da análise intervalar tradicional que, além da teoria de conjuntos, utiliza a lógica proposicional;
- Definição de novos operadores (menos conservadores) para computar o *regret* no IRMOEA-MR. Por exemplo, pode-se definir um operador que ao invés de retornar um valor único para cada solução, retorne um conjunto de valores de acordo com número de subpavimentos n_i. Matematicamente pode ser definido por:

$$\mathbf{R} = \begin{Bmatrix} \left([\mathrm{f}]_1(\mathbf{x},[\mathbf{p}]_1)^+ - \mathrm{v}_1(\mathbf{V}_1), \ldots, [\mathrm{f}]_k(\mathbf{x},[\mathbf{p}]_1)^+ - \mathrm{v}_k(\mathbf{V}_1), \ldots, [\mathrm{f}]_{n_f}(\mathbf{x},[\mathbf{p}]_1)^+ - \mathrm{v}_{n_f}(\mathbf{V}_1)\right), \ldots, \\ \left([\mathrm{f}]_1(\mathbf{x},[\mathbf{p}]_\mathrm{j})^+ - \mathrm{v}_1(\mathbf{V}_\mathrm{j}), \ldots, [\mathrm{f}]_k(\mathbf{x},[\mathbf{p}]_\mathrm{j})^+ - \mathrm{v}_k(\mathbf{V}_\mathrm{j}), \ldots, [\mathrm{f}]_{n_f}(\mathbf{x},[\mathbf{p}]_\mathrm{j})^+ - \mathrm{v}_{n_f}(\mathbf{V}_\mathrm{j})\right), \ldots, \\ \left([\mathrm{f}]_1(\mathbf{x},[\mathbf{p}]_{n_i})^+ - \mathrm{v}_1(\mathbf{V}_{n_i}), \ldots, [\mathrm{f}]_k(\mathbf{x},[\mathbf{p}]_{n_i})^+ - \mathrm{v}_k(\mathbf{V}_{n_i}), \ldots, [\mathrm{f}]_{n_f}(\mathbf{x},[\mathbf{p}]_{n_i})^+ - \mathrm{v}_{n_f}(\mathbf{V}_{n_i})\right) \end{Bmatrix}. \quad (9\text{-}1)$$

REFERÊNCIAS BIBLIOGRÁFICAS

ALOTTO, P.; BAUMGARTNER, U.; FRESCHI, F.; JAINDL, M.; KOSTINGER, A.; MAGELE, C.; RENHART, W.; REPETTO, M. SMES Optimization Benchmark Extended: Introducing Pareto Optimal Solutions Into TEAM 22. **IEEE Transactions on Magnetics**, v. 44, n. 6, p. 1066-1069, june 2008.

ALOTTO, P.; CAITI, A.; MOLINARI, G.; REPETTO, M. A Multiquadrics-based Algorithm for the Acceleration of Simulated Annealing Optimization Procedures. **IEEE Transactions on Magnetics**, v. 32, n. 3, p. 1198-1201, May 1996.

ALOTTO, P.; KUNTSEVICH, A. V.; MAGELE, C.; MOLINARI, G.; PAUL, C.; REPETTO, M.; RICHTER, K. Multiobjective Optimization in Magnetostatics: A Proposal for a Benchmark Problem. **IEEE Transactions on Magnetics**, v. 32, n. 3, p. 1238-1241, May 1996.

ASTROM, K. J.; HAGGLUND, T. The Future of PID Control. **Control Engineering Practice**, v. 9, n. 11, p. 1163-1175, November 2001.

ASTROM, K. J.; HAGGLUND, T. Revisiting the Ziegler-Nichols Step Response Method for PID Control. **Journal of Process Control**, v. 14, n. 6, p. 635-650, September 2004.

AVIGAD, G.; BRANKE, J. **Embedded Evolutionary Multi-objective Optimization for Worst Case Robustness**. Genetic and Evolutionary Computation Conference - GECCO'08. Atlanta: ACM. 2008. p. 617-624.

BACK, T.; HAMMEL, U.; SCHWEFEL, H. P. Evolutionary Computation: comments on the History and Current State. **IEEE Transactions On Evolutionary Computation**, v. 1, n. 1, p. 3-17, April 1997.

BARBOSA, C. H. N. R. **Reconfiguração e Restauração Ótima de Sistemas de Distribuição Primária de Energia Elétrica**. UFMG - Tese de Doutorado. Belo Horizonte, p. 128. 2012.

BARRICO, C.; ANTUNES, C. An Evolutionary Approach for Assessing the Degree of Robustness of Solutions to Multi-Objective Models. **Evolutionary Computation in Dynamic and Uncertain Environments. Series: Studies in Computational Intelligence**, Berlin-Heidelberg, v. 51, p. 565–582, 2007.

BEALE, M. H.; HAGAN, M. T.; DEMUTH, H. **Neural Network Toolbox User's Guide R2011b**. Natick: MathWorks Incorporation, 2011.

BEN-TAL, A.; EL GHAOUI, L.; NEMIROVSKI, A. **Robust Optimization**. Princeton and Oxford: Princeton University Press, 2009.

BEYER, H. G.; SENDHOFF, B. Robust Optimization - A Comprehensive Survey. **Computer Methods in Applied Mechanics and Engineering**, v. 196, n. 33-34, p. 3190-3218, July 2007.

BRAGA, A. P.; CARVALHO, A. P. L. F.; LUDERMIR, T. B. **Redes Neurais Artificiais:** Teoria e Aplicações. 2. ed. Rio de Janeiro: LTC, 2007.

BRANKE, J. Creating Robust Solutions by Means of Evolutionary Algorithms. **Lecture Notes in Computer Science**, Amsterdam, v. 1498, p. 119-128, September 1998.

BROOMHEAD, D. S.; LOWE, D. Multivariable Functional Interpolation and Adaptive Networks. **Complex Systems**, v. 2, n. 3, p. 321-355, 1988.

BURKILL, J. C. **Functions of Intervals**. Proceedings of the London Mathematical Society. London: LMS. 1924. p. 275-310.

BYRD, R. H.; LU, P.; NOCEDAL, J.; ZHU, C. A Limited Memory Algorithm for Bound Constrained Optimization. **SIAM Journal on Scientific Computing**, Philadelphia, v. 16, n. 5, p. 1190-1208, September 1995.

COELLO, C. A. **Handling Preferences in Evolutionary Multiobjective Optimization:** A Survey. Proceedings of the 2000 Congress on Evolutionary Computation. San Diego: IEEE. 2000. p. 30-37.

COELLO, C. A. A Evolutionary Multi-Objective Optimization: a Historical View of the Field. **IEEE Computational Intellligence Magazine**, v. 1, n. 1, p. 28-36, February 2006.

COELLO, C. A.; CHRISTIANSEN, A. D. Two New GA-based Methods for Multiobjective Optimization. **Civil Engineering and Environmental Systems**, v. 15, n. 3, p. 207-243, 1998.

COELLO, C. A.; LAMONT, G.; VELDHUIZEN, D. **Evolutionary Algorithms for Solving Multi-Objective Problems**. 2. ed. New York: Springer, 2007.

DARWIN, C. **On the Origin of Species by Means of Natural Selection**. London: John Murray, 1859.

DEB, K. **Multi-objective Genetic Algorithms: Problem Difficults and Construction of Test Problems**. University of Dortmund. Dortmund, p. Technical Report CI-49. 1998.

DEB, K.; AGRAWAL, R. B. Simulated Binary Crossover for Continuous Search Space. **Complex Systems**, v. 9, n. 2, p. 115-148, April 1995.

DEB, K.; GUPTA, H. Introducing Robustness in Multi-objective Optimization. **Evolutionary Computation**, v. 14, n. 4, p. 463-494, November 2006.

DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. **IEEE Transactions on Evolutionary Computation**, v. 6, n. 2, p. 182-197, April 2002.

DIAS, A. H. F. **Algoritmos Genéticos Aplicados a Problemas com Múltiplos Objetivos**. UFMG - Dissertação de Mestrado. Belo Horizonte. 2000.

DIAS, A. H. F.; VASCONCELOS, J. A. Multiobjective Genetic Algorithms Applied to Solve Optimization Problems. **IEEE Transactions on Magnetics**, v. 38, n. 2, p. 1133-1136, March 2002.

EIBEN, A. E.; HINTERDING, R.; MICHALEWICZ, Z. Parameter Control in Evolutionary Algorithms. **IEEE Transactins on Evolutionary Computation**, v. 3, n. 2, p. 124-141, July 1999.

FONSECA, C. M.; FLEMING, P. **On the Perfomance Assessment and Comparision of Stochastic Multiobjective optimizers**. Parallel Problem Solving from Nature - PPSN IV. Berlim: Springer-Verlag. 1996. p. 584-593.

FONSECA, C. M.; FLEMING, P. J. **Genetic Algorithms for Multiobjective Optimization:** Formulation, Discussion and Generalization. Proceedings of the Fifth International Conference on Genetic Algorithms. San Mateo: Morgan Kaufmann. 1993. p. 416-423.

FONSECA, C. M.; FLEMING, P. J. An Overview of Evolutionary Algorithms in Multiobjective Optimization. **Evolutionary Computation**, Sheffield, v. 3, n. 1, p. 1-16, April 1995.

FONSECA, C. M.; FONSECA, V. G.; PAQUETE, L. **Exploring the Performance of Stochastic Multiobjective Optimisers with the Second-Order Attainment Function**. Evolutionary Multi-Criterion Optimization. Guanajuato: Springer. 2005. p. 250-264.

FONSECA, V. G.; FONSECA, C. M.; HALL, A. O. **Inferential Performance Assessment of Stochastic Optimisers and the Attainment Function**. First International Conference on Evolutionary Multi-Criterion Optimization. Zurich: Springer-Verlag. 2001. p. 213-225.

GAING, Z. A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System. **IEEE Trans. Energy Conversion**, v. 19, n. 2, p. 384-391, June 2004.

GOH, C. K.; TAN, K. C. Evolving the Tradeoffs between Pareto-Optimality and Robustness. **Evolutionary Computation in Dynamic and Uncertain Environments. Series: Studies in Computational Intelligence**, Berlin - Heidelberg, v. 51, p. 457-478, 2007.

GOH, C. K.; TAN, K. C. **Evolutionary Multi-objective Optimization in Uncertain Environments**. 1. ed. Berlin Heidelberg: Springer, 2009.

GOH, C. K.; TAN, K. C.; CHEONG, C. Y.; ONG, Y. S. **Noise-Induced Features in Robust Multi-Objective Optimization Problems**. IEEE Congress on Evolutionary Computation. Singapore: IEEE. 2007. p. 568-575.

GORISSEN, D. **Grid-enabled Adaptive Surrogate Modeling for Computer Aided Engineering**. University of Antwerp - Thesis. Antwerp. 2010.

GUIMARÃES, F. G.; LOWTHER, D. A.; RAMÍREZ, J. A. Multiobjective Approaches for Robust Electromagnetic Design. **IEEE Transactions on Magnetics**, v. 42, n. 4, p. 1207-1210, April 2006.

GUIMARÃES, F. G.; PINTO, F. C. F.; SALDANHA, R. R.; IGARASHI, H.; RAMÍREZ, J. A. A Multi-objective Proposal for the Team Benchmark Problem 22. **IEEE Transactions on Magnetics**, v. 42, n. 4, p. 1471-1474, April 2006.

GUPTA, S.; ROSENHEAD, J. Robustness in Sequential Investment Decisions. **Management Science**, v. 15, n. 2, p. 18-29, October 1968.

HANSEN, M.; JASZKIEWICZ, A. **Evaluating the Quality of Approximation to the Non-dominated Set**. Technical University of Denmark. Lyngby, p. 30, IMM Technical Report. 1998.

HICKEY, T.; JU, Q.; EMDEN, M. H. Interval Arithmetic: From Principles to Implementation. **Journal of the Association for Computing Machinery**, New York, v. 48, n. 5, p. 1038-1068, September 2001.

HO, S. L.; YANG, S. A Population-Based Incremental Learning Method for Robust Optimal Solutions. **IEEE Transactions on Magnetics**, v. 46, n. 8, p. 3189-3192, August 2010.

HO, S.; YANG, S.; NI, G.; CHENG, K. An Efficient Tabu Search Algorithm for Robust Solutions of Eletromagnetic Design Problems. **IEEE Transactions on Magnetics**, v. 44, n. 6, p. 1042-1045, June 2008.

HORN, J.; NAFPLIOTIS, N. **Multiobjective Optimization Using the Niched Pareto Genetic Algorithms**. University of Illinois. Urbana-Champaign, p. 32, Technical Report no. 93005. 1993.

HOWARD, D.; ROBERTS, S. C. **Genetic Programming Solution of the Convection-difusion Equation**. Genetic and Evolutionary Computation Conference. San Francisco: Morgan Kaufmann. 2001. p. 34-41.

JAULIN, L.; KIEFFER, M.; DIDRIT, O.; WALTER, E. **Applied Interval Analysis with Examples in Parameter and State Estimation, Robust Control and Robotics**. 1. ed. London: Springer, 2001.

JIN, Y.; BRANKE, J. Evolutionary Optimization in Uncertain Environments - A Survey. **IEEE Transactions on Evolutionary Computation**, v. 9, n. 3, p. 303-317, June 2005.

JINHUA, Z.; JIAN, Z.; HAIFENG, D.; SUN'AN, W. Self-organizing Genetic Algorithm Based Tuning of PID Controllers. **Information Sciences**, v. 179, n. 7, p. 1007-1018, March 2009.

KEIJZER, M. **Improving Symbolic Regression with Interval Arithmetic and Linear Scaling**. Genetic Programming Proceedings of EuroGP'2003. Essex: Springer-Verlag. 2003. p. 71-83.

KEIJZER, M. Scaled Symbolic Regression. **Genetic Programming and Evolvable Machines**, Hingham, v. 5, n. 3, p. 259-269, September 2004.

KNOWLES, J. D. **Local Search and Hybrid Evolutionary Algorithms for Pareto Optimization**. University of Reading - Thesis. Reading. 2002.

KOUVELIS, P.; YU, G. **Robust Discret Optimization and its Application**. 1. ed. Dordrecht: Kluwer Academic Publishers, 1997.

KOZA, J. **Genetic Programming - On the Programming of Computers by Means of Natural Selection**. 1. ed. Cambridge: MIT Press, 1992.

LEBENSZTAJN, L.; MARRETTO, C. A. R.; COSTA, M. C.; COULOMB, J. -L. Kriging: A Useful Tool for Electromagnetic Device Optimization. **IEEE Transactions on Magnetics**, v. 40, n. 2, p. 1196-1199, March 2004.

LEE, K.; PARK, G. Robust Optimization Considering Tolerances of Design Variables. **Computers & Structures**, v. 79, n. 1, p. 77-86, January 2001.

LI, Y.; ANG, K. H.; CHONG, G. C. Y. PID Control System Analysis and Design. **IEEE Control Systems Magazine**, v. 26, n. 1, p. 32-41, February 2006.

LÓPEZ-IBÁNEZ, M.; STUTZLE, T.; PAQUETE, L. **Ghaphical Tools for the Analysis of Bi-objective Optimization Algorithms**. GECCO workshop on Theoretical Aspects of Evolutionary Multiobjective Optimization. Portland: ACM. 2010. p. 1959-1962.

MATTSON, C. A.; MESSAC, A. Pareto Frontier Based Concept Selection Under Uncertainty, with Visualization. **Optimization and Engineering**, v. 6, n. 1, p. 85-115, March 2005.

MENDES, M. H. S.; COSME, L. B.; CAMINHAS, W. M.; VASCONCELOS, J. A. **Aplicação de Modelagem Nebulosa em Problemas de Otimização sob Condições de Incertezas**. 1º Congresso Brasileiro de Sistemas Fuzzy – CBSF-2010. Sorocaba: Anais do CBSF. 2010. p. 463-470.

MENDES, M. H. S.; SOARES, G. L.; COULOMB, J. L.; VASCONCELOS, J. A. **Appraisal of Surrogate Modeling Techniques for Electromagnetic Device**. 15th Biennial IEEE Conference on Electromagnetic Field Computation - CEFC 2012. Oita - Japan: IEEE. 2012. p. 79.

MENDES, M. H. S.; SOARES, G. L.; COULOMB, J. L.; VASCONCELOS, J. A. **Comparison of Surrogate Modeling Approaches on TEAM Workshop Problem 22**. 15º SBMO – Simpósio Brasileiro de Micro-ondas e Optoeletrônica e o 10º CBMag – Congresso Brasileiro de Eletromagnetismo - MOMAG 2012. João Pessoa: Anais do MOMAG em CD, Artigo # 100638_1. 2012.

MENDES, M. H. S.; SOARES, G. L.; COULOMB, J. L.; VASCONCELOS, J. A. **Surrogate Model Determination by Using Genetic Programming**. 15th Biennial IEEE Conference on Electromagnetic Field Computation - CEFC 2012. Oita - Japan: IEEE. 2012. p. 76.

MENDES, M. H. S.; SOARES, G. L.; VASCONCELOS, J. A. PID Step Response Using Genetic Programming. **Lecture Notes in Computer Science**, Berlin, Heidelberg, v. 6457, p. 359-368, 2010.

MIETTINEN, K. M. **Nonlinear Multiobjective Optimization**. 1. ed. Boston, London, Dordrecht: Kluwer Academic Publishers, 1999.

MIETTINEN, K. M.; DEB, K.; JAHN, J.; OGRYCZAK, W.; SHIMOYAMA, K.; VETSCHERA, R. Future Challenges. **Lecture Notes in Computer Science**, v. 5252, p. 435-461, 2008.

MOORE, R. E. **Interval Arithmetic and Automatic Error Analysis in Digital Computing**. Stanford University - Thesis. Stanford. 1962.

MOORE, R. E. **Interval Analysis**. 1. ed. New Jersey: Prentice-Hall Englewood Cliffs, 1966.

MOORE, R. E.; KEARFOOT, R. B.; CLOUD, M. J. **Introduction to Interval Analysis**. 1. ed. Philadelphia: SIAM, 2009.

NEUMAIER, A. Improving Interval Enclosures. **Reliable Computing**, v. To appear, p. 1-28, 2009.

OGATA, K. **Modern Control Engineering**. 3. ed. Englewood Cliffs: Prentice Hall, 1997.

OLIVEIRA DA SILVA, S. G. **Controlling Bloat: Individual and Population Based Approaches in Genetic Programming**. Universidade de Coimbra - Thesis. Coimbra. 2008.

OLIVEIRA, P. W.; DIVERIO, T. A.; CLAUDIO, D. M. **Fundamentos da Matemática Intervalar**. 1. ed. Porto Alegre: Sagra Luzzatto, 2005.

ONG, Y.; PRASANTH, B.; LUM, K. Max-Min Surrogate-Assisted Evolutionary Algorithm for Robust Design. **IEEE Transactions on Evolutionary Computation**, v. 10, n. 4, p. 392-404, August 2006.

PAENKE, I.; BRANKE, J.; JIN, Y. Efficient Search for Robust Solutions by Means of Evolutionary Algorithms and Fitness Approximation. **IEEE Transactions on Evolutionary Computation**, v. 10, n. 4, p. 405-420, August 2006.

PARREIRAS, R. O. **Algoritmos Evolucionários e Técnicas de Tomada de Decisão em Análise Multicritério**. UFMG - Tese de Doutorado. Belo Horizonte. 2006.

PARREIRAS, R. O.; VASCONCELOS, J. A. Decision Making in Multiobjective Optimization Aided by the Multicriteria Tournament Decision Method. **Nonlinear Analysis**, v. 71, n. 12, p. 191-198, December 2009.

PEDERSEN, G. K. M.; YANG, Z. **Multi-objective PID-controller Tuning for a Magnetic Levitation System Using NSGA-II**. Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation. Seattle: ACM. 2006. p. 1737-1744.

PEDRYCZ, W.; EKEL, P.; PARREIRAS, R. **Models and Algorithms of Fuzzy Multicriteria Decision-Making and Their Applications**. 1. ed. United Kingdom: John Wiley & Sons, 2011.

PENNACHIN, C. L.; LOOKS, M.; VASCONCELOS, J. A. **Robust Symbolic Regression with Affine Arithmetic**. Genetic and Evolutionary Computation Conference. Portland: ACM. 2010. p. 917-924.

PERNY, P.; SPANJAARD, O.; STORME, L. A Decision-theoretic Approach to Robust Optimization in Multivalued Graphs. **Annals of Operations Research**, v. 147, n. 1, p. 317-341, October 2006.

PFLUGFELDER, D.; WILKENS, J.; OELFKE, U. Worst Case Optimization: a Method to Accoutn for Uncertainties in the Optimization of Intensity Modulated Proton Therapy. **Physics in Medicine and Biology**, v. 53, n. 6, p. 1689-1700, March 2008.

POLI, R.; LANGDON, W. B.; MCPHEE, N. F. **A Field Guide to Genetic Programming**. 1. ed. : Published at http://www.gp-field-guide.org.uk, 2008.

ROCCO, C. M.; SALAZAR, D. E. A Hybrid Approach Based on Evolutionary Strategies and Interval Arithmetic to Perform Robust Designs. **Evolutionary Computation in Dynamic and Uncertain Environments. Series: Studies in Computational Intelligence**, v. 51, p. 543-564, 2007.

ROSENHEAD, J.; ELTON, M.; GUPTA, S. Robustness and Optimality as Criteria for Strategic Decisions. **Operational Research Quaterly**, v. 23, n. 4, p. 413-430, December 1972.

RUETSCH, G. R. An Interval Algorithm for Multi-Objective Optimization. **Structural and Multidisciplinary Optimization**, v. 30, n. 1, p. 27-37, July 2005.

RUMP, S. M. INTLAB: INTerval laboratory. **Developments in Reliable Computing**, Dordrecht, p. 77-104, 1999.

RUSTEM, B.; HOWE, M. **Algorithms for Worst-Case Design and Applications to Risk Management**. 1. ed. London and New Jersey: Princeton University Press, 2002.

SCHOTT, J. R. **Fault Tolerant Design Using Single and Multi-criteria Genetic Algorithms**. Massachusetts Institute of Technology - Thesis (M.S.). Boston. 1995.

SMITS, G.; KONTANCHEK, M. Pareto Front Exploitation in Symbolic Regression. In: O'REILLY, U. M., et al. **Genetic Programming Theory and Practice II**. New York: Springer, 2004. p. 283-300.

SOARES, G. L. **Algoritmos Genéticos: Estudo, Novas Técnicas e Aplicações**. UFMG - Dissertação de Mestrado. Belo Horizonte. 1997.

SOARES, G. L. **Algoritmos Determinístico e Evolucionário Intervalares para Otimização Robusta Multi-Objetivo**. UFMG - Tese de Doutorado. Belo Horizonte. 2008.

SOARES, G. L.; ADRIANO, R. L. S.; MAIA, C. A.; JAULIN, L.; VASCONCELOS, J. A. Robust Multi-Objective TEAM 22 Problem: A Case Study of Uncertainties in Design Optimization. **IEEE Transactions on Magnetics**, v. 45, n. 3, p. 1028-1031, March 2009.

SOARES, G. L.; GUIMARÃES, F. G.; MAIA, C. A.; VASCONCELOS, J. A.; JAULIN, L. **Interval Robust Multi-Objective Evolutionary Algorithm**. IEEE Congress on Evolutionary Computation. Trondheim - Norway: IEEE. 2009. p. 1637-1643.

SOARES, G. L.; PARREIRAS, R. O.; JAULIN, L.; VASCONCELOS, J. A.; MAIA, C. A. Interval Robust Multi-objective Algorithm. **Nonlinear Analysis: Theory, Methods & Applications**, v. 71, n. 12, p. 1818-1825, December 2009.

SRINIVAS, N.; DEB, K. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. **Evolutionary Computation**, v. 2, n. 3, p. 221-248, 1994.

SUNAGA, T. Theory of an Interval Algebra and its Application to Numerical Analysis. **RAAG Memoirs**, v. 2, p. 29-46, 1958.

TAGUCHI, G. **Quality Engineering Through Design Optimization**. New York: Kraus International Publications, 1984.

TAKAHASHI, R. H. C.; PERES, P. L. D.; FERREIRA, P. A. V. Multiobjective H2/H∞ Guaranteed Cost PID Design. **IEEE Control Systems Magazine**, v. 17, n. 5, p. 37-47, October 1997.

TILBURY, D.; LUNTZ, J.; MESSNER, W. **Controls Education on the WWW:** Tutorial for Matlab and Simulink. Proceedings of the American Control Conference. Philadelphia: IEEE. 1998. p. 1304-1308.

TSUTSUI, S.; GHOSH, A. Genetic Algorithms with a Robust Solution Searching Scheme. **IEEE Transactions on Evolutionary Computation**, v. 1, n. 3, p. 201-208, September 1997.

TSUTSUI, S.; GHOSH, A.; FUJIMOTO, Y. **A Robust Solution Searching Scheme in Genetic Search**. Parallel Problem Solving from Nature. Berlin: Springer. 1996. p. 543-552.

VASCONCELOS, J. A.; RAMIREZ, J. A.; TAKAHASHI, R. H. C.; SALDANHA, R. R. Improvements in Genetic Algorithms. **IEEE Transactions on Magnetics**, September, v. 37, n. 5, p. 3414-3417, 2001.

VELDHUIZEN, D. A.; LAMONT, G. B. **Multiobjective Evolutionary Algorithm Test Suites**. Proceedings of the 1999 ACM symposium on Applied computing. San Antonio: ACM. 1999. p. 351-357.

VINCKE, P. About Robustness Analysis. **Newsletter of the European Working Group "Multicriteria Aid for Decisions"**, v. 3, n. 8, p. 7-9, 2003.

VLADISLAVLEVA, E. L.; SMITS, G. F.; HERTOG, D. Order of Nonlinearity as a Complexity Measure for Models Generated by Simbolic Regression via Pareto Genetic Programming. **IEEE Transactions on Evolutionary Computation**, v. 13, n. 2, p. 333-349, April 2009.

WHITLEY, D. An Overview of Evolutionary Algorithms: Practical Issues and Common Pitfalls. **Information and Software Technology**, December, v. 43, n. 14, p. 817-831, 2001.

YANG, S.; ONG, Y.; JIN, Y. **Evolutionary Computation in Dynamic and Uncertain Environments - Series:** Studies in Computational Intelligence. 1. ed. Berlin-Heidelberg: Springer, v. 51, 2007.

YOUNG, R. C. The Algebra of Many-Valued Quantities. **Mathematische Annalen**, v. 104, n. 1, p. 260-290, 1931.

ZANG, T.; HEMSCH, M.; HILBURGER, M.; KENNY, S.; LUCKRING, J.; MAGHAMI, P.; PADULA, S.; STROUD, W. **Needs and Opportunities for Uncertainty-Based Multidisciplinary Design Methods for Aerospace Vehicles**. Technical Memorandum - National Aeronautics and Space Administration - NASA. Washington, p. 54. 2002.

ZITZLER, E.; LAUMANNS, M.; THIELE, L. **Spea2: Improving the Strength Pareto Evolutionary Algorithm**. Technical Report-103 - Swiss Federal Institute of Technology. Zurich, p. 21. 2001.

ZITZLER, E.; THIELE, L. **Multiobjective Optimization Using Evolutionary Algorithms - a Comparative Case Study**. Proceedings of 5th International Conference on Parallel Problem Solving from Nature. Amsterdam: Springer. 1998. p. 292-301.

ZITZLER, E.; THIELE, L. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. **IEEE Transactions on Evolutionary Computation**, v. 3, n. 4, p. 257-271, November 1999.

ZITZLER, E.; THIELE, L.; LAUMANNS, M.; FONSECA, C. M.; FONSECA, V. G. Performance Assessment of Multiobjective Optimizers: an Analysis and Review. **IEEE Transactions on Evolutionary Computation**, v. 7, n. 2, p. 117-132, April 2003.