

**Universidade Federal de Minas Gerais**  
Programa de Pós-Graduação em Engenharia Elétrica

**École Nationale Supérieure d'Ingénieurs**

Laboratoire Extraction et Exploitation  
de l'Information en Environnements Incertains

**Université de Bretagne Occidentale**

# T E S E

submetida à banca examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais como parte dos requisitos necessários para obtenção do grau de

**Doutor em Engenharia Elétrica**

**Área: OTIMIZAÇÃO**

Apresentada e defendida por

**Gustavo Luís SOARES**

**Algoritmos Determinístico e Evolucionário Intervalares  
para Otimização Robusta Multi-Objetivo**

Tese elaborada em regime de co-tutela entre UFMG, ENSIETA e UBO.

## **Orientadores**

<i>Brasil</i>	João Antônio de VASCONCELOS	-	UFMG
	Carlos Andrey MAIA	-	UFMG
<i>França</i>	Luc JAULIN	-	ENSIETA

Apresentada em 24 outubro de 2008 na UFMG.

---

## FICHA TÉCNICA

### Instituições

Universidade Federal de Minas Gerais  
École Nationale Supérieure d'Ingénieurs  
Université de Bretagne Occidentale

### Título

Algoritmos Determinístico e Evolucionário Intervalares  
para Otimização Robusta Multi-Objetivo

### Autor

Gustavo Luís SOARES

### Orientação

<i>Orientador (Brasil)</i>	João Antônio de VASCONCELOS	- UFMG
<i>Co-orientador (Brasil)</i>	Carlos Andrey MAIA	- UFMG
<i>Orientador (França)</i>	Luc JAULIN	- ENSIETA

### Banca Examinadora

<i>Presidente</i>	Laurent KRÄHENBÜHL	- ECL
<i>Revisores</i>	Laurent KRÄHENBÜHL	- ECL
	Petr Iakovlevitch EKEL	- PUC-MG
<i>Examinadores</i>	João Antônio de VASCONCELOS	- UFMG
	Carlos Andrey MAIA	- UFMG
	Luc JAULIN	- ENSIETA
	Laurent HARDOUIN	- LISA
<i>Convidados</i>	Reinaldo Martinez PALHARES	- UFMG
	Douglas Alexandre Gomes VIEIRA	- UFMG
	Frederico Gadelha GUIMARÃES	- UFOP

---

## AGRADECIMENTOS

*“Mas o Senhor esteve a meu lado  
como um forte guerreiro ...”, Jeremias 20, 11.*

Juntamente com Ele, muitas pessoas cruzaram o meu caminho me incentivando e colaborando para o desenvolvimento de meu trabalho. Portanto, venho prestar-lhes minha sincera gratidão.

Aos meus orientadores João Antônio de Vasconcelos, Carlos Andrey Maia e Luc Jaulin pelo tempo a mim investido, pelo conhecimento transmitido, pela disponibilidade, paciência e amizade.

Aos meus professores na pós-graduação Renato Cardoso Mesquita, Jaime Arturo Ramirez, Ricardo Hiroshi Caldeira Takahashi, Walmir Matos Caminhas e João Antônio de Vasconcelos, pela disponibilidade, pelas contribuições preciosas na definição de meu tema de trabalho e pelas oportunidades profissionais que surgiram a partir de suas disciplinas e do convívio com eles.

Aos coordenadores e secretárias do PPGEE da UFMG, Benjamim Rodrigues de Menezes, Hani Camille Yehia, Anete Vieira e Arlete Vidal de Freitas, pelo auxílio nas questões formais do programa de pós-graduação.

À secretária Annick Billot-Coat, ao Patrick Leon, na época responsável pelos estágios e parcerias internacionais, e ao Professor Luc Jaulin, pela ajuda nas questões formais para realização do estágio na ENSIETA.

Aos colegas Andreas Arnold-Bos, Ricardo Luiz Adriano e Roberta Oliveira Parreiras, pela colaboração científica que resultou nos artigos publicados no período dessa tese.

Aos colegas Gledson Melotti e Tales Argolo Jesus, pela colaboração científica que resultou no desenvolvimento do Capítulo 5 dessa tese.

Aos colegas do GOPAC, Alexandre Ramos Fonseca, Douglas Alexandre Gomes Vieira, Luciano Cunha de Araújo Pimenta, Frederico Gadelha Guimarães, pela integração, motivação e pelo suporte técnico nos momentos iniciais de meu trabalho.

Aos meus amigos na França, Othmane Benhmammouch, Yasar Kutuvan-tavida, Lionel Cros e Ludivine Laval, pela acolhida, pelas aulas de francês e inglês, pelo suporte na parte informal do meu estágio doutoral.

À Sra. Elani Vieira Soares e ao Sr. Geraldo Belarmino Soares (pais), e à Cristiana Silva de Miranda Souto Soares (esposa) pelo suporte, carinho, paciência e motivação.

À Sra. Iredes Maria Silva Souto (sogra), ao Sr. José de Miranda Souto (sogro), à Elenice Teixeira Chaves (irmã), ao Marcos Antônio Carvalho Chaves (cunhado), pelos diversos tipos ajuda que me prestaram.

A todos meus amigos, parentes e funcionários, pela paciência e compreensão em minhas incontáveis ausências.

#### *Agradecimento Especial*

A meu irmão, amigo e sócio Gilberto Augusto Soares, por todas as vezes que realizou minhas tarefas em nossas empresas, pelo apoio, pela paciência em aguardar a finalização de meu trabalho.

*Com amor e carinho ...*

*a Sra. Elani, Sr. Geraldo,  
Cristiana, Sofia e Enzo.*

## RESUMO

Esta Tese considera a presença de incertezas na modelagem do problema de otimização não linear multi-objetivo. Para solucionar esse problema de otimização robusta, três métodos de busca originais foram propostos utilizando a filosofia do pior caso, isto é, a metodologia na qual se busca determinar as variáveis de projeto eficientes para o pior caso das incertezas. Os métodos propostos são o [I]RMOA I e [I]RMOA II, que utilizam técnicas intervalares, e o [I]RMOEA que utiliza algoritmos evolucionários e técnicas intervalares. As soluções eficientes encontradas neste contexto, são denominadas de soluções robustas eficientes. Esses algoritmos são descritos detalhadamente, bem como a análise de suas características, vantagens e desvantagens. Além destas contribuições originais, este trabalho também propõe: a) uma técnica de nichos, baseada em intervalos, útil para manter diversidade nas populações em algoritmos evolucionários; b) uma métrica para medir a uniformidade da distribuição de pontos eficientes; c) um conjunto de funções teste, adaptadas para otimização robusta; e d) a descrição e resolução de um problema de otimização robusta multi-objetivo envolvendo sintonia de controladores PID. Em adição, este texto apresenta e discute as fontes de incertezas, as interpretações probabilísticas e determinísticas da interferência das incertezas no sistema de otimização.

## RÉSUMÉ

Cette thèse traite du problème d'optimisation non linéaire multiobjectif avec des paramètres d'optimisation non parfaitement connus. Afin de résoudre ce problème d'optimisation robuste non linéaire multiobjectif, trois méthodes de recherche sont proposées dans cette thèse. Les deux premières méthodes utilisent l'analyse d'intervalle et sont baptisées [I]RMOA I et [I]RMOA II. La troisième méthode, nommée [I]RMOEA, utilise quant à elle, des algorithmes évolutionnaires couplés à l'analyse d'intervalle. Les solutions efficaces trouvées dans ce contexte sont appelées solutions robustes efficaces. Une description détaillée de ces algorithmes est fournie dans ce manuscrit en proposant une analyse de leurs caractéristiques, leurs avantages et leurs inconvénients. En complément des contributions innovantes précédentes, cette thèse propose par ailleurs : a) une technique de niche basée sur l'analyse d'intervalle, utile pour maintenir la diversité dans les populations des algorithmes évolutionnaires, b) une métrique permettant de mesurer l'uniformité de la distribution des points efficaces, c) un ensemble de fonctions test appropriées à l'optimisation robuste, d) la description et la résolution d'un problème d'optimisation robuste multiobjectif concret portant sur la synthèse de contrôleurs PID. Enfin, ce document présente les sources d'incertitudes et fournit les interprétations probabilistes et déterministes de l'interférence de celle-ci dans le système d'optimisation multiobjectif.

## ABSTRACT

This thesis considers the presence of uncertainties in the modeling of nonlinear multi-objective optimization problem. To solve these nonlinear multi-objective robust optimization problems, three original search methods has been proposed using worst-case scenario strategies: [I]RMOA I and [I]RMOA II, that use interval deterministic techniques, and [I]RMOEA that uses evolutionary algorithms and deterministic interval techniques. The found efficient solutions in this context are called efficient robust solutions or non-dominated robust solutions. These algorithms are at great length described, as well as the analysis of their characteristics, advantages and disadvantages. Beyond these original contributions, this work also considers: a) one technique of niches, based on intervals, useful to keep diversity in the populations in evolutionary algorithms; b) a metric technique to measure uniformity in the non-dominated points distribution; c) a set of test functions, suitable for robust optimization; e d) the description and resolution of a multi-objective robust optimization problem about PID controller tuning. In addition, this text presents and argues the sources of uncertainties, as well as their probabilistic and deterministic interpretations in optimization problems.

---

## CONTRIBUIÇÕES

As contribuições apresentadas ao longo dessa tese, consideradas como destaques, foram classificadas nos tópicos descritos a seguir.

### Publicações

[2008] **Soares, G. L.**, Parreiras, R. O., Jaulin, L., Maia, C. A. and Vasconcelos, J. A., “Interval Robust Multi-Objective Algorithm”, World Congress of Nonlinear Analysts WCNA 2008, Orlando, Florida, July, 2008.

[2008] **Soares, G. L.**, Boss, A. A., Jaulin, L. and Maia, C. A. and Vasconcelos, J. A., An Interval-Based Target Tracking Approach for Range-Only Multistatic Radar, IEEE Transactions on Magnetics, vol. 44, issue 6, pg.1350-1353, 2008.

[2008] **Soares, G. L.**, Adriano, R. L., Maia, C. A., Jaulin, L. and Vasconcelos, J. A., Robust Multi-Objective TEAM 22 Problem: a Case Study of Uncertainties in Design Optimization, 13th Biennial IEEE Conference on Electromagnetic Field Computation CEFC 2008, Athens, Greece, May, 2008.

[2005] **Soares, G. L.**, Vasconcelos, J. A. e Maia, C. A., Algoritmo Evolucionário por Eleições, 1º Seminário do Programa de Pós-Graduação em Engenharia Elétrica, Belo Horizonte, Brasil, 2005.

### Conceitos

O Capítulo “Algoritmos Intervalares e Evolucionário-Intervalar para Otimização Robusta Multi-Objetivo” apresenta muitos aspectos que foram considerados como contribuições importantes, como por exemplo: a) o confronto de interpretações do termo “robusto” por diversos autores (texto introdutório do capítulo); b) a formulação robusta multi-objetivo restrita, com a definição do conjunto solução robusto e da fronteira de Pareto robusta (Seção 3.3); c) a classificação das regiões no espaço dos objetivos em “acima” da fronteira robusta e “abaixo” da fronteira robusta (Seção 3.3); e d) os métodos intervalares para otimização robusta, para classificação de nichos e para medir desempenho de algoritmos multi-objetivo (Seções 3.4-3.7).

## Métodos

- [I]RMOA I(Interval Robust Multi-Objective Algorithm), Seção 3.4
- [I]RMOA II(Interval Robust Multi-Objective Algorithm II), Seção 3.5
- [I]RMOEA(Interval Robust Multi-Objective Evolutionary Algorithm), Seção 3.6
- NB[I] (Nichos por Bisseção Intervalar), Subseção 3.6.2
- MB[I] (Métrica por Bisseção Intervalar), Subseção 3.7.1

## Aplicação

No capítulo destinado à resolução de um problema real via algoritmos propostos, foi escolhido como tema a otimização de sintonia de controladores PID. De fato, existem trabalhos científicos que tratam deste problema. A novidade está na abordagem robusta multi-objetiva e restrita apresentada e na resolução do problema através dos algoritmos propostos nesta Tese. Foram desenvolvidas duas formulações: a) uma geral (5.16), onde o pesquisador interessado pode utilizá-la para definir seu problema específico; b) um formulação detalhada (5.22) para um estudo de caso específico.

## Outras contribuições

Essa Tese contribui para o Programa de Pós-Graduação em Engenharia da Universidade Federal de Minas Gerais através da introdução de duas linhas de pesquisa ainda não investigadas por outros trabalhos desenvolvidos no programa. São elas: Otimização Robusta Multi-Objetivo e Análise Intervalar.

## NOTAÇÃO

A notação empregada foi baseada em padrões de representação normalmente encontrados nas literaturas matemática e de engenharia. Entretanto, mesmo nestas áreas, é comum encontrar trabalhos científicos utilizando notações distintas para denotar, por exemplo, um mesmo parâmetro, grandeza ou operação matemática. Nestes casos, foram escolhidos padrões de nomenclatura que pudessem também atender à simbologia adotada para os parâmetros específicos deste trabalho. Em alguns casos, o leitor pode discordar quanto à escolha da notação. Justifica-se que as definições e escolhas de símbolos não se deram por facilidade e conveniência, e sim pela tentativa de descrever os algoritmos, formulação e outras partes do texto de forma clara, objetiva e não ambígua. Infelizmente, devido à grande quantidade de símbolos, talvez o leitor considere que as metas acima não tenham sido atingidas. Os critérios de notação e algumas nomenclaturas básicas empregados neste texto estão descritos logo abaixo. Uma lista mais completa dos símbolos com sua descrição está apresentada nas próximas páginas desse trabalho. Registra-se que essa lista não cobre os parâmetros e constantes que auxiliam de forma local a explicação de equações, métodos e processos considerados óbvios e desnecessários.

Como de costume,  $\mathbb{R}$  é utilizado para representar o conjunto dos números reais;  $\mathbb{IR}$ , os números intervalares reais;  $\mathbb{Z}$ , os números inteiros; e  $\mathbb{IB}$ , o conjunto booleano intervalar. O uso de expoentes numéricos nesses símbolos denota a ordem do espaço matemático que eles representam, por exemplo  $\mathbb{IR}^2$ . Outros conjuntos, que representam parte dos conjuntos anteriores, são apresentados com letras maiúsculas em negrito, como em **A**. Essa mesma notação é empregada para matrizes. O contexto descarta a possível confusão entre conjuntos e matrizes. O negrito em letras minúsculas denota vetores como é o caso de  $\mathbf{x}$ , e listas encadeadas de vetores como em  $\mathbf{Q}_{[\mathbf{x}]}$ , que significa uma estrutura de dados tipo fila cujos elementos são vetores intervalares. O acesso aos elementos de listas encadeadas é feito com o uso de chaves, como em  $\mathbf{Q}_{[\mathbf{x}]} \{1\}$ . Todas as estruturas de dados do tipo fila são iniciadas por **Q** (Queue) e as estruturas do tipo pilha, por **S** (Stack).

Os termos em inglês estão em parênteses e em fonte normal. Os termos em latim, em itálico. O itálico também é reservado para apresentação inicial de termos técnicos dentro do texto corrente. Vocábulo ou trechos em destaque no texto ou que representam uma linguagem menos formal se encontram entre aspas. Os intervalos são delimitados por colchetes, como na função de inclusão  $[f]$ . Os limites inferiores de intervalos são denotados pelo acréscimo de grifos acima e abaixo símbolo, como em  $\underline{f}_i$  e  $\overline{f}_i$ . As letras  $i$ ,  $j$ ,  $k$  e  $t$  denotam iterações de processos, identificam elementos de conjuntos e par-

---

particularizam termos de vetores, entre outros usos. Por exemplo,  $f_{j\mathbf{x}^i}$  significa o  $j$ -ésimo termo da função objetivo  $\mathbf{f}$  quando avalia o  $i$ -ésimo indivíduo  $\mathbf{x}^i$  de uma população. O expoente ' é utilizado como uma notação suplementar ou de transformação do termo assinalado. Por exemplo,  $\mathbf{X}$  é o espaço de busca;  $\mathbf{X}'$  é o espaço de busca após a exclusão da região não viável. O expoente \* indica a informação de otimalidade da base que o acompanha, como  $\mathbf{X}^*$  simboliza o conjunto de soluções do problema de otimização.

Finalmente, fugindo um pouco dos padrões, algumas expressões literais foram acrescentadas ao vocabulário técnico para explicar melhor dado processo. Por exemplo, examine o uso dos termos “estreita” e “envelope” no seguinte texto:

*Funções intervalares em  $\mathbb{R}$  resultam em dois números; um para o limite inferior e outro para o limite superior. Uma função intervalar “estreita” significa função intervalar onde os limites inferior e superior estão próximos. Além disso, um “envelope” dessa mesma função intervalar significa uma região no espaço das funções que contém intervalo resposta da função intervalar.*

Assim, após a descrição de termos novos como feito acima ou por meio de definição formal, considerou-se que uma frase que utilize os verbos “estretar” e “envelopar”, ou outras formas similares dos termos, podem ser empregadas no texto corrente com significado técnico.

Provavelmente, existam exceções que fujam às regras apresentadas. Nestes casos, espera-se que o contexto e o rigor das definições sejam suficientes para deixar claras quaisquer outras classes de nomenclatura não cobertos aqui ou na lista completa apresentada no final deste trabalho.

## NOMENCLATURA

### Romanos

- $\mathbf{c}([\mathbf{x}])$  Centro de caixa intervalar
- $C(s)$  Função de transferência do controlador
- $\mathbf{des}_t$  Desempenho da população na geração  $t$
- $\mathbf{des}'_t$  Desempenho útil da população na geração  $t$
- $e_i$  Contador de soluções Pareto - métrica TE
- $e(t)$  Sinal de erro do sistema ou diferença entre a referência e a saída do sistema
- $E(s)$  Sinal de erro do sistema no domínio da frequência
- $\mathbf{f}$  Vetor de funções objetivo - notação genérica e simplificada
- $f_{front}^i$  Desempenho de um indivíduo  $i$  segundo a fronteira que ele participa
- $f_{nicho}^i$  Desempenho de um indivíduo  $i$  segundo o nicho que ele participa
- $\mathbf{f}(\mathbf{x})$  Vetor de funções objetivo - problema de otimização tradicional
- $\mathbf{f}(\mathbf{x}^*)$  Imagem da solução ótima ou ponto da fronteira de Pareto - problema de otimização tradicional
- $\mathbf{f}(\mathbf{X})$  Imagem direta do espaço de busca, espaço dos objetivos
- $\mathbf{f}(\mathbf{X}^*)$  Fronteira de Pareto, imagem do conjunto solução  $\mathbf{X}^*$  - problema de otimização tradicional e robusta
- $\mathbf{f}(\mathbf{x}, \varepsilon)$  Vetor de funções objetivo com incerteza na medição de parte dos argumentos de  $\mathbf{f}$
- $\mathbf{f}(\mathbf{x}, \xi)$  Vetor de funções objetivo com incerteza devido ao ambiente computada
- $\mathbf{f}([\mathbf{x}])$  Função extensão intervalar da função  $\mathbf{f}(\mathbf{x})$ , vetor de funções intervalares, vetor funções objetivo com argumento intervalar
- $\mathbf{f}(\mathbf{x}, \mathbf{p})$  Vetor de funções objetivo - problema de otimização robusta
- $\mathbf{f}(\mathbf{X}', \mathbf{P})$  Imagem do espaço de busca viável sujeito às incertezas do sistema
- $[\mathbf{f}]$  Função inclusão para objetivos - notação genérica e simplificada
- $[\mathbf{f}]^*$  Função de inclusão mínima - notação genérica e simplificada
- $[\mathbf{f}]_{[\mathbf{x}]}^*$  Função de inclusão mínima para a caixa/indivíduo  $[\mathbf{x}]$  considerando  $[\mathbf{P}]$

---

$[\mathbf{f}]_{\mathbf{x}}^*$	Função de inclusão mínima para o ponto/indivíduo $\mathbf{x}$ considerando $[\mathbf{P}]$
$[\mathbf{f}]([\mathbf{x}])$	Função de inclusão da função $\mathbf{f}([\mathbf{x}])$
$[\mathbf{f}](\mathbf{pop}_t, [\mathbf{P}])$	Função objetivo para cálculo do desempenho da população na geração $t$
$[\mathbf{F}]$	Nicho corrente
$[\mathbf{F}]_0$	Nicho inicial que contém toda a população
$\mathbf{g}(\mathbf{x})$	Vetor de funções de restrição - problema de otimização tradicional
$\mathbf{g}([\mathbf{x}])$	Vetor de funções de restrição - problema de otimização intervalar
$\mathbf{g}(\mathbf{x}, \mathbf{p})$	Vetor de funções de restrição - problema de otimização robusta
$[\mathbf{g}]$	Função de inclusão de restrições - notação genérica e simplificada
$[\mathbf{g}]([\mathbf{x}], [\mathbf{p}])$	Função de inclusão de restrições - problema de otimização robusta
$G(s)$	Função de transferência resultante
$H(s)$	Função de transferência da planta
HV	Hipervolume de um conjunto solução - métrica do hipervolume
HVR	Hipervolume normalizado de um conjunto solução - métrica do hipervolume
$K_d$	Ganho proporcional
$K_i$	Ganho integral
$K_p$	Ganho proporcional
$\mathbf{K}$	Vetor de parâmetros do PID
$\mathbf{K}^-$	Representação computável de $\mathbf{Y}^-$
$\mathbf{K}^+$	Representação computável de $\mathbf{Y}^+$
$\mathbf{K}^{**}$	Solução robusta de projeto, selecionada na tomada de decisão
$m(t)$	Resposta do controlador
$M_p$	Sobre-sinal máximo
$n_{bis}$	Número de bissecções
$n_{bits}$	Número de bits utilizado para codificação dos indivíduos
$n_{gen}$	Número máximo de gerações
$n_{nicho}$	Número de integrantes de dado nicho
$n_{pop}$	Número de indivíduos da população
$n_{sp}$	Número de subpavimentos
$p_{1-4}$	Polinômios de Kharitonov

---

$\mathbf{p}$	Parâmetro de incerteza geral - representa $\epsilon$ , $\xi$ e $\varepsilon$
$\mathbf{P}$	Espaço de incertezas
$\mathbf{P}'$	Espaço de incertezas finito
$[\mathbf{p}]$	Parâmetro de incerteza
$[\mathbf{P}]$	Espaço de incerteza
$p_c$	Probabilidade de cruzamento
$p_m$	Probabilidade de mutação
$\text{pop}_t$	População na geração $t$
$\text{proj}_{\mathbf{A}}(\mathbf{Z})$	Projeção do conjunto $\mathbf{Z}$ sobre o conjunto $\mathbf{A}$
$\mathbf{Q}_{\mathbf{F}}$	Lista FIFO com nichos ativos
$\mathbf{Q}_{\mathbf{K}^-}$	Lista FIFO pontos que definem $\mathbf{K}^-$
$\mathbf{Q}_{\mathbf{K}^+}$	Lista FIFO pontos que definem $\mathbf{K}^+$
$\mathbf{Q}_{\mathbf{Y}^*}$	Lista FIFO pontos ideais para fronteira Pareto robusta
$\mathbf{Q}_{\text{fxp}}$	Lista FIFO com amostras avaliadas da função grade
$\mathbf{Q}_{\text{ind}\{\cdot\}}$	Lista FIFO com índices dos indivíduos de $\mathbf{Q}_{[\mathbf{F}]\{\cdot\}}$
$\mathbf{Q}_{\text{ind}}^{[\mathbf{F}]}$	Lista FIFO com índices dos indivíduos de $\mathbf{Q}_{[\mathbf{F}]\{1\}}$
$\mathbf{Q}_{[\mathbf{F}]}^v$	Lista FIFO com índices dos indivíduos de são vértices de caixas que separam soluções
$\mathbf{Q}_{[\mathbf{p}]}$	Lista FIFO com subpavimentos do espaço de parâmetros
$\mathbf{Q}_{[\mathbf{x}]}$	Lista FIFO com subpavimentos do espaço de busca
$r(t)$	Referência ou entrada do sistema
$r_{bis}$	Rodada de bissecção
$r_{bis}^{max}$	Número máximo de rodadas de bissecção
$R(s)$	Referência do sistema no domínio da frequência
$s$	Variável de Laplace
$\mathbf{S}_{[\mathbf{x}]}$	Lista LIFO com subpavimentos do espaço de busca
$\mathbf{S}_{[\mathbf{p}]}$	Lista LIFO com subpavimentos do parâmetro de incerteza
$t$	Período de tempo em que o sistema é observado
$t_a$	Tempo de acomodação
$\mathbf{t}(\mathbf{x})$	Teste de inclusão real

---

$[t]([\mathbf{x}])$	Teste de inclusão intervalar
$t_p$	Tempo de pico
$t_s$	Tempo de subida
$\mathbf{u}^{\max}$	Ponto ideal para maximização
$\mathbf{u}^{\min}$	Ponto ideal para minimização
$v([\mathbf{x}])$	Volume de caixa intervalar
$w([\mathbf{x}])$	Largura de caixa intervalar
$\mathbf{x}$	Parâmetro ou variável de busca, ou de projeto
$\tilde{\mathbf{x}}$	Parâmetro de busca impreciso, por incerteza na medição, na construção e em arredondamentos
$\mathbf{x}^*$	Solução ótima ou eficiente - problema de otimização tradicional e robusta
$\mathbf{x}_0^*$	Solução ótima - exemplo
$\bar{\mathbf{x}}$	Limite superior do intervalo $[\mathbf{x}]$
$\underline{\mathbf{x}}$	Limite inferior do intervalo $[\mathbf{x}]$
$\mathbf{X}$	Espaço de busca ou das variáveis de busca
$\mathbf{X}'$	Espaço de busca viável
$\mathbf{X}^*$	Conjunto solução ou de pontos eficientes - problema de otimização tradicional e robusta
$\mathbf{X}_d^*$	Conjunto solução discreto
$\mathbf{X}_s^*$	Conjunto solução encontrado ao fim de um processo de otimização
$[\mathbf{x}]$	Intervalo de busca
$\mathbf{x}^i$	Indivíduo $i$ de uma dada população
$[\mathbf{x}]_1, [\mathbf{x}]_2$	Caixas resultantes da bissecção de $[\mathbf{x}]$
$\mathbf{Y}^*$	Fronteira de Pareto - problema de otimização tradicional e robusta
$\mathbf{y}$	Imagem de $\mathbf{f}(\mathbf{x}, \mathbf{p})$
$\mathbf{Y}'$	Imagem do espaço de busca viável problema otimização tradicional e robusta
$\mathbf{Y}^+$	Porção acima da fronteira robusta
$\mathbf{Y}^-$	Porção abaixo da fronteira robusta
$[\mathbf{Y}']$	Imagem computada do espaço de busca viável em forma intervalar
$\mathbf{Y}_d^*$	Fronteira de Pareto robusta discreta, imagem do conjunto solução discreto

---

$\mathbf{Y}_s^*$	Fronteira de Pareto robusta ou imagem do conjunto solução quando produtos do processo de otimização
$[z]$	Função de inclusão para resposta do sistema de controle $z$
$Z(s)$	Resposta do sistema de controle no domínio da frequência
$z(t)$	Saída do sistema de controle

### Gregos e similares

$\alpha$	Variável para simplificação, definição em (5.19)
$\alpha'$	Variável para simplificação, definição em (5.21)
$\xi$	Parâmetro de incerteza - representa a influência do ambiente
$\mathcal{C}$	Nome de métrica usada para comparar conjuntos não dominados
$\Delta$	Métrica do espalhamento
$\Delta\mathbf{K}^+$	Porção de $\mathbf{Y}^+$ não computável diretamente
$\Delta\mathbf{K}^-$	Porção de $\mathbf{Y}^-$ não computável diretamente
$\partial\mathbf{K}^+$	Fronteira inferior de $\mathbf{K}^+$
$\partial\mathbf{K}^-$	Fronteira superior de $\mathbf{K}^-$
$\epsilon$	Parâmetro de incerteza - tolerância, erro máximo aceitável para $\tilde{\mathbf{x}}$
$\varepsilon$	Parâmetro de incerteza - imprecisão na medição de $\mathbf{f}$
$\varepsilon_{[\mathbf{x}]}$	Precisão para $[\mathbf{x}]$
$\varepsilon_{\mathbf{x}\#}$	Precisão para a função grade
$\varepsilon_{TE}$	Precisão para a métrica TE
$\mathcal{S}$	Métrica do espaçamento
$\sigma_{v([\mathbf{F}])}$	Desvio padrão das caixas entre soluções - Técnica MB[I]
$\Upsilon_{\leq}$	Representa o subconjunto com os menores os pontos do conjunto que aparece como argumento
$\Upsilon_{\geq}$	Representa o subconjunto com os maiores os pontos do conjunto que aparece como argumento

### Expoentes

$n_f$	Ordem, dimensão - espaço dos objetivos
$n_g$	Ordem, dimensão - espaço das restrições
$n_n$	Ordem, dimensão - espaço qualquer
$n_p$	Ordem, dimensão - espaço de incertezas

$n_x$  Ordem, dimensão - espaço de busca

## Operadores

- Fim de teoremas e provas
- \ Complemento de conjuntos
- | | Número de elementos do conjunto apresentado entre os símbolos |
- ◇ Operador binário que representa as operações +, -, \* e /
- ⊖ Operador subtração para tratar a dependência na computação de intervalos
- ⊘ Operador divisão para tratar a dependência na computação de intervalos
- ⋈ Operador de dominância, indicando que o operando da esquerda é estritamente inferior ao da direita; o símbolo pode ser usado entre números reais, entre números reais e intervalos e entre intervalos
- ⋉ Operador de dominância, indicando que o operando da direita é estritamente inferior ao da esquerda; o símbolo pode ser usado entre números reais, entre números reais e intervalos e entre intervalos
- ⋊ Operador de dominância, indicando que o operando da esquerda tem valor numérico inferior ao da direita; o símbolo pode ser usado entre números reais, entre números reais e intervalos e entre intervalos
- ⋋ Operador de dominância, indicando que o operando da direita tem valor numérico inferior ao da esquerda; o símbolo pode ser usado entre números reais, entre números reais e intervalos e entre intervalos
- ⋌ Operador de dominância, indicando que o operando da esquerda não tem valor numérico inferior ao da direita; o símbolo pode ser usado entre números reais, entre números reais e intervalos e entre intervalos
- ⋍<sub>C</sub> Indica que operando da esquerda é completamente melhor que o da direita
- ⋍<sub>F</sub> Indica que operando da esquerda é fortemente melhor que o da direita
- ⋍<sub>f</sub> Indica que operando da esquerda é fracamente melhor que o da direita

## Algoritmos de otimização

[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

CO-MOEA Coevolutionary Multi-Objective Evolutionary Algorithm

DRMOGA Divided Range Multi-Objective Genetic Algorithm

ENORA Evolutionary Algorithm of Non Dominated Sorting with Radial Slots

MμGA Multi-objective MicroGA

MOGA Multi-Objective Genetic Algorithm  
NPGA Niche Pareto Genetic Algorithm  
NSGA Non-Sorting Genetic Algorithm  
NSGA-II Non-Sorting Genetic Algorithm II  
PAES Pareto Archived Evolution Strategy  
PSO Particle Swarm Optimization  
SGA Simple Genetic Algorithm  
SPEA Strength Pareto Evolutionary Algorithm  
VEGA Vector Evaluated Genetic Algorithm  
RMOGA Robust Multi-Objective Genetic Algorithm  
WBGGA Weight-Based Genetic Algorithm

### **Acrônimos**

CSC Computable Sufficient Conditions  
EAs Evolutionary Algorithms  
EEA Election Evolutionary Algorithm  
FIFO First In First Out - lista encadeada tipo fila  
FON Função Fonseca e Fleming  
FPS Feasible Point Searcher  
GA Algoritmos Genéticos  
ICP Interval Constraint Propagation  
ISE Integral Square-Error  
LIFO Last In First Out - lista encadeada tipo pilha  
MB[I] Métrica por Bisseção Intervalar  
MB[I]<sub>front</sub> Valor conhecido da métrica MB[I] na fronteira  
MB[I]<sub>max</sub> Valor máximo computado da métrica MB[I]  
MB[I]<sub>norm</sub> Valor normalizado da métrica MB[I]  
MIF Minimal Inclusion Function  
MOEA Multi-objective Evolutionary Algorithms  
NB[I] Nichos por Bisseção Intervalar  
RH Critério de estabilidade de Routh-Hurwitz

SCH2 Função Schaffer 2

SIVIA Set Inversion Via Interval Analysis

SMES Superconducting Magnetic Energy Storage System

TE Taxa de Erro

ZDT1 Função 1 da família de funções de Zitzler, Deb e Thiele

ZDT2 Função 2 da família de funções de Zitzler, Deb e Thiele

ZDT3 Função 3 da família de funções de Zitzler, Deb e Thiele

# CONTEÚDO

<i>Parte I Tese em Francês - Resumo</i>	1
<i>Parte II Tese em Português</i>	57
1. <i>Introdução</i> . . . . .	58
1.1 Justificativa . . . . .	59
1.2 Objetivos . . . . .	61
1.2.1 Geral . . . . .	61
1.2.2 Específicos . . . . .	61
1.3 Escopo . . . . .	62
1.4 Metodologia . . . . .	63
1.5 Organização do Trabalho . . . . .	64
1.6 Considerações Finais . . . . .	65
2. <i>Revisão Bibliográfica</i> . . . . .	66
2.1 Otimização Multi-Objetivo . . . . .	66
2.1.1 Fundamentos e Definições . . . . .	67
2.1.2 Problema de Otimização Multi-Objetivo . . . . .	68
2.1.3 Classificação dos Métodos Multi-Objetivos . . . . .	69
2.2 Algoritmos Evolucionários Multi-Objetivos . . . . .	71
2.2.1 Estrutura Geral dos Algoritmos Evolucionários . . . . .	72
2.2.2 Algoritmos Evolucionários para Problemas Multi-Objetivo . . . . .	75
2.2.3 Diversidade de Algoritmos Evolucionários . . . . .	77
2.2.4 Parâmetros Gerais . . . . .	79
2.2.5 Elitismo . . . . .	80
2.2.6 Métricas para Medir Desempenho . . . . .	81
2.2.7 Técnicas de Nicho . . . . .	86
2.3 Análise Intervalar . . . . .	87
2.3.1 Breve Histórico . . . . .	88
2.3.2 Relações Algébricas de Conjuntos . . . . .	89
2.3.3 Fundamentos e Definições . . . . .	90
2.3.4 Operações Aritméticas Elementares . . . . .	91

---

2.3.5	Funções Intervalares Elementares . . . . .	93
2.3.6	Dependência . . . . .	94
2.3.7	Vetores Intervalares . . . . .	96
2.3.8	Funções de Inclusão . . . . .	97
2.3.9	Lógica Booleana Intervalar . . . . .	100
2.3.10	Teste de Inclusão . . . . .	101
2.3.11	Subpavimentos . . . . .	101
2.3.12	SIVIA . . . . .	102
2.3.13	Inequações Intervalares . . . . .	103
2.4	Otimização Multi-objetivo Intervalar . . . . .	104
2.5	Considerações Finais . . . . .	107
3.	<i>Algoritmos para Otimização Robusta Multi-Objetivo</i> . . . . .	109
3.1	Breve Histórico da Otimização Robusta . . . . .	110
3.2	Conceitos e Notações . . . . .	111
3.2.1	Comparação de vetores reais e intervalares . . . . .	111
3.3	Problema Multi-Objetivo Robusto . . . . .	113
3.4	Algoritmo Intervalar Multi-Objetivo Robusto I . . . . .	116
3.4.1	Preliminares . . . . .	116
3.4.2	[I]RMOA I . . . . .	118
3.5	Algoritmo Intervalar Multi-Objetivo Robusto II . . . . .	122
3.5.1	Preliminares . . . . .	122
3.5.2	[I]RMOA II . . . . .	126
3.6	Algoritmo Evolucionário Intervalar Multi-Objetivo Robusto . . . . .	126
3.6.1	Preliminares . . . . .	127
3.6.2	Técnica de Nicho por Bissecção Intervalar . . . . .	129
3.6.3	[I]RMOEA . . . . .	133
3.7	Métricas para Medir Desempenho Revisadas . . . . .	135
3.7.1	Métrica por Bissecção Intervalar - MB[I] . . . . .	136
3.7.2	MB[I] × Métricas Apresentadas . . . . .	139
3.8	Considerações Finais . . . . .	144
4.	<i>Validação Numérica dos Algoritmos para Otimização Robustos</i> . . . . .	147
4.1	Metodologia . . . . .	148
4.2	Funções Teste . . . . .	148
4.3	Testes e Resultados . . . . .	155
4.3.1	[I]RMOA I . . . . .	156
4.3.2	[I]RMOA II . . . . .	168
4.3.3	[I]RMOEA . . . . .	179
4.4	Testes Complementares . . . . .	197
4.4.1	[I]RMOA I como Ferramenta de Validação . . . . .	197

---

4.4.2	[I]RMOA II × Dimensionalidade do Espaço de Busca . . . . .	200
4.4.3	[I]RMOEA × Dimensionalidade do Espaço de Busca . . . . .	201
4.5	Considerações Finais . . . . .	202
5.	<i>Otimização Robusta Multi-Objetivo para Sintonia de Controladores</i>	
	<i>PID</i> . . . . .	205
5.1	Revisão sobre o Controlador PID . . . . .	206
5.2	Descrição do Problema . . . . .	207
5.2.1	Análise da Estabilidade do Sistema . . . . .	208
5.2.2	Formulação do Problema Multi-Objetivo Robusto para Sistemas de Controle com Planta Intervalar . . . . .	210
5.3	Estudo de Caso . . . . .	212
5.3.1	Caracterização da Planta . . . . .	213
5.3.2	Resposta Temporal do Sistema . . . . .	213
5.3.3	Formulação do Problema Multi-Objetivo Robusto . . . . .	214
5.3.4	Funções Objetivo . . . . .	215
5.3.5	Análise das Funções Objetivo e de Restrição . . . . .	216
5.3.6	Experimento e Resultados . . . . .	218
5.4	Considerações Finais . . . . .	226
6.	<i>Conclusões e Trabalhos Futuros</i> . . . . .	228
6.1	Conclusões . . . . .	228
6.2	Trabalhos Futuros . . . . .	231
	<i>Apêndice</i> . . . . .	241
	<i>A. Equações para os Cálculos do Tempo de Pico e do Sobre-Sinal Máximo</i>	242
	<i>B. Equação para o Tempo de Subida</i> . . . . .	243
	<i>C. Equações para o ISE</i> . . . . .	244

Parte I

RÉSUMÉ EN FRANÇAIS DE LA THÈSE

---

## *Préface au Résumé Français*

Ce résumé suit la structure du rapport de la thèse. Les sections de chaque chapitre sont décrites dans ce résumé en utilisant les expressions mathématiques et les tableaux les plus importants de ce rapport, lorsque les figures restent en portugais. Le titre intitulé “Contribuições” (page ix), dans les pages préliminaires du rapport, contient la liste des contributions du travail de recherche comme les articles publiés, les formulations proposées et les algorithmes développés, par exemple. De plus, au début des chapitres, en “Notações” (page xi), sont décrits les différentes notations mathématiques utilisées pour identifier les variables, les ensembles mathématiques, les opérations, les structures de données, ainsi que les autres types de paramètres qui ont méritent des spécifications standards pour mieux les distinguer dans le texte.

Le premier chapitre présente comment les sources des incertitudes associent, de manière générale, à un système d’optimisation. Il est également présenté comment établir une formulation multi-objectif robuste à partir d’une formulation classique, ce qui constitue l’objectif principal répondu dans cette thèse. Ce premier chapitre conclue par traiter de l’importance de ce sujet de thèse, les objectifs et les méthodologies de recherche.

Le chapitre deux fournit un état de l’art en recherche qui a été la base pour la modélisation des algorithmes développés. Il présente des langages issus des communautés d’Optimisation Multi-Objectif, d’Algorithmes Évolutionnaires et d’Analyse par Intervalles qui simplifient la compréhension et la construction des algorithmes proposés pour l’optimisation robuste.

Le troisième chapitre présente d’abord les concepts de base qui permettent comprendre les algorithmes développés dans ce rapport. Les algorithmes sont décrits en détail par la suite. De plus, une technique de niche par intervalles est proposée et des métriques pour évaluer les résultats trouvés par les méthodes.

Cinq fonctions tests ont été adaptées de l’optimisation multi-objectif classique pour l’optimisation robuste. Les résultats de tests sont présentés et analysés dans le chapitre quatre et permettent de valider les performances des algorithmes d’optimisation multi-objectif.

Le chapitre cinq présente une application expérimentale permettant d’expliquer comment les méthodes proposées peuvent être appliquer à un problème complet. Ainsi, pas à pas, une étude de cas d’un problème réel a été réalisée dès la conception à la résolution. À travers ce cas, ont été montrés des avantages des algorithmes et mis en évidence quelques difficultés techniques rencontrées durant la réalisation du processus d’optimisation.

Enfin, les conclusions et les perspectives de ce rapport constitue le dernier chapitre de ce rapport.

## Chapitre 1 - Introduction

Cette section présente un problème d'optimisation multi-objectif à travers d'un exemple pour démarrer les discussions diverses sur la présence des incertitudes en systèmes d'optimisation multi-objectif. L'exemple en question se formule tel que :

$$\left\{ \begin{array}{l} \min_{\mathbf{x} \in \mathbb{R}^2} \quad f_1(\mathbf{x}) = (x_1 - c_1)^2 + (x_2 - c_2)^2; \\ \quad \quad \quad f_2(\mathbf{x}) = c_3x_1 - c_4x_2. \\ \\ s.c. : \\ \quad \quad \quad g(\mathbf{x}) = x_1 + x_2 \leq 0. \end{array} \right. \quad (1.1)$$

où  $x_1$  et  $x_2$  sont les variables réelles,  $c_1$  à  $c_4$  sont coefficients positives,  $g(x)$  est la fonction de contrainte à satisfaire et  $f_1(x_1, x_2)$  et  $f_2(x_1, x_2)$  sont les objectifs à optimiser. Une méthode pour résoudre un problème comme (1.1) consiste à chercher un ensemble de solutions, appelées solutions Pareto, qui sont non dominées entre elles et possèdent une distribution uniforme dans le Front de Pareto. Les caractéristiques de non dominance et de distribution sont mesurées dans l'espace des objectifs.

En analysant (1.1), il est clair que ce système d'optimisation n'est pas considéré en présence d'incertitudes sur les paramètres, qui sont présents dans beaucoup problèmes d'optimisation. Ensuite, d'autres scénarios ont été imaginés dans lesquels les incertitudes existent autant pour mesurer et pour construire  $x_1$  et  $x_2$  que des imprecisions pour quantifier les coefficients  $c_i$ ,  $i = 1 \dots 4$ . Ainsi, ce nouveau scénario défini par (1.1) est devenu inconsistant, et les solutions du problème selon la formulation (1.1) qui ont été trouvées ne sont peut être plus valides. Ainsi, l'expression (1.1) a besoin d'être redéfinie par une formulation appelée *robuste*, pour caractériser le scénario avec des incertitudes et offrant des conditions aux méthodes pour trouver les solutions optimales.

### 1.1 - L'importance du sujet de thèse

Les problèmes réels d'optimisation nécessitent souvent de la prise en compte de plusieurs critères. Formellement, un système d'optimisation multi-objectif traditionnel sans contraintes est souvent défini comme :

$$\min_{\mathbf{x} \in \mathbb{R}^{n_x}} \mathbf{f}(\mathbf{x}) \quad (1.2)$$

où  $\mathbf{x} \in \mathbb{R}^{n_x}$  est le vecteur des variables du problème, et  $\mathbf{f}(\mathbf{x}) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_f}$  est le vecteur des objectifs à optimiser.

Ensuite, on considère que ce système a été affecté par différents types d'incertitudes. Par exemple, des incertitudes peuvent exister de part les imprécisions sur les paramètres du modèle et de part les caractéristiques physiques environnementales du milieu (pression, température, viscosité, etc). Par conséquent, il y a en général plusieurs possibilités pour modéliser un système d'optimisation qui prennent en compte des incertitudes. Cette thèse utilise un vecteur de paramètre  $\mathbf{p}$  pour représenter tous les sortes d'incertitudes dans l'expression du vecteur de fonctions objectif, noté par  $\mathbf{f}(\mathbf{x}, \mathbf{p})$ . Ainsi, en considérant  $\epsilon$  et  $\varepsilon$  les incertitudes d'imprécision de mesurer et de quantifier  $\mathbf{x}$  et  $\mathbf{f}$ , et  $\xi$  le facteur d'ambient, la formulation robuste de  $\mathbf{f}(\mathbf{x})$  est définie par :

$$\mathbf{f}(\mathbf{x}, \mathbf{p}) = \mathbf{f}(\mathbf{x}, \epsilon, \varepsilon, \xi), \quad \mathbf{p} \in \mathbb{R}^{n_p}. \quad (1.6)$$

où  $\mathbf{p} = (\mathbf{x}, \epsilon, \varepsilon, \xi)$ .

De plus, d'après le paragraphe précédent, on peut écrire  $\mathbf{f}(\mathbf{x}) \neq \mathbf{f}(\mathbf{x}, \mathbf{p})$ . En conséquence, les résultats d'optimisation de  $\mathbf{f}(\mathbf{x})$  et  $\mathbf{f}(\mathbf{x}, \mathbf{p})$  peuvent être aussi différents. En principe, il n'est pas possible de déduire l'influence de  $\mathbf{p}$  sur  $\mathbf{f}(\mathbf{x}, \mathbf{p})$ , sauf dans les cas triviaux. Donc, si les incertitudes sont présentes dans le problème d'optimisation, alors les solutions trouvées utilisant  $\mathbf{f}(\mathbf{x})$  peuvent perdre l'attribut optimal.

Il y a plusieurs études dans la littérature sur les méthodes pour résoudre les problèmes multi-objectif traditionnelles comme (1.2). Mais, il y n'existe que peu de travaux qui traite d'optimisation avec incertitudes. Par conséquent, nous avons jugé nécessaire de considérer les problèmes multi-critères avec incertitudes et de développer des méthodes pour chercher cette sorte de solutions, dit solutions robustes.

## 1.2 - L'objectifs

En considérant le vecteur des paramètres de projet  $\mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^{n_x}$  et le paramètre d'incertitude  $\mathbf{p} \in \mathbf{P} \subseteq \mathbb{R}^{n_p}$ . Le but principal de cette thèse consiste à étudier et à résoudre des problèmes d'optimisation multi-objectif robustes définis comme :

$$\begin{aligned} \min_{\mathbf{x} \in \mathbf{X}} \quad & \max_{\mathbf{p} \in \mathbf{P}} \mathbf{f}(\mathbf{x}, \mathbf{p}) \\ \text{s.c. :} \quad & \mathbf{g}(\mathbf{x}, \mathbf{p}) \leq \mathbf{0}, \quad \mathbf{p} \in \mathbf{P}, \end{aligned} \quad (1.7)$$

---

où  $\mathbf{f}(\mathbf{x}, \mathbf{p}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_f}$  est le vecteur des fonctions objectif, et  $\mathbf{g}(\mathbf{x}, \mathbf{p}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_g}$  est le vecteur des fonctions de contraintes.

À partir de ces études, trois algorithmes multi-critères ont été proposés pour résoudre le problème (1.7). Ces algorithmes ont été développés en utilisant des techniques basées sur les méthodes intervalles et évolutionnaires.

Les objectifs secondaires sont le suivants :

- réaliser une étude de l'état de l'art sur la théorie de l'Optimisation Multi-Objectif et l'Analyse par Intervalles mettant en évidence les principales caractéristiques des algorithmes ;
- présenter d'autres philosophies sur l'optimisation robuste et situer la formulation adoptée dans ce document ;
- développer des techniques par intervalles pour les procédés internes aux algorithmes proposés ;
- proposer cinq fonctions test pour l'optimisation robuste multi-objectif ;
- valider les algorithmes grâce à ces fonctions test ;
- développer un problème multi-objectif réel et appliquer les algorithmes développés pour le résoudre.

### 1.3 - Limitations

Il y a plusieurs façons de traiter et d'interpréter l'optimisation robuste, et aussi de développer les algorithmes de résolution. Donc, il a eu besoin d'établir quelques limitations ce travail. La thématique générale de cette thèse a été limitée à :

- employer l'optimisation seulement au format de minimisation - la maximisation peut être obtenu en remplaçant  $\mathbf{f}(\mathbf{x})$  par  $\mathbf{f}(\mathbf{x}, \mathbf{p})$  ;
- utiliser seulement intervalles réels - les intervalles complexes n'ont pas été traités ;
- définir un intervalle comme un nombre incertain qui existe dans les limites inférieures et supérieures d'un intervalle ;

- présenter seulement les concepts fondamentaux d'Optimisation Multi-Objectif, Algorithmes Évolutionnaires Multi-Objectif et Analyse par Intervalles concernant les notions importantes pour le développement des algorithmes proposés - des études plus profondes sur les sujets n'ont pas été réalisées ;
- développer des problèmes test en utilisant opérations et fonctions élémentaires comment  $+$ ,  $-$ ,  $\times$ ,  $\div$ ,  $\exp$ ,  $\sin$ ,  $\cos$ , etc - la bibliothèque mathématique des intervalles codifiés pendant la thèse ne contient pas opérateurs et fonctions plus complexes ;
- développer algorithmes que n'utilise pas l'information de dérivées dans le processus de recherche - les algorithmes que utilisent ce genre d'information sont abondants dans la littérature ;
- utiliser la structure des algorithmes génétiques pour élaborer la structure d'algorithme évolutionnaire hybride proposé - d'autres sortes d'algorithmes évolutionnaires ne seront pas utilisées ;
- utiliser les méthodes traditionnelles pour la sélection, mutation et croisement - la diversité de ces méthodes est grand et donc impossible d'analyser ;
- utiliser la codification binaire pour représenter les paramètres de l' algorithme évolutionnaire proposé.

## 1.4 - Méthodologie

Dans cette section, il a été expliqué, comment cette étude a été développée. Les pas ont été :

- l'étude d'état de l'art sur les sujets de la recherche ;
- développement d'une librairie des codes sources pour effectuer les opérations avec intervalles ;
- construction des standards, définitions et du modèle du Problème d'Optimisation Robuste ;
- développement des algorithmes ;
- élaboration des fonctions test ;

- application des algorithmes développés dans un problème réel.

## 1.5 - Planification

Le Chapitre 2 présente les fondements théoriques sur l'Optimisation Multi-Objectif, l'Algorithmes Évolutionnaires et l'Analyse par Intervalles, indispensables au développement des algorithmes présentés au Chapitre 3. Au-delà de les méthodes pour l'optimisation robuste, le Chapitre 3 introduit d'autres techniques par intervalles comme une technique de niche et une métrique pour mesurer la distribution de points dans de la Frontière de Pareto. Dans le Chapitre 4, des problèmes tests sont proposés afin de valider les algorithmes. Plusieurs tests sont faits. La formulation et la résolution d'un problème multi-objectif pour la synthèse des contrôleurs PID ont été faites au Chapitre 5. Finalement, sur le Chapitre 6, les conclusions sont présentées.

## Chapitre 2 - Révision Bibliographique

### 2.1 - Optimisation Multi-Objectif

Dans cette section plusieurs questions d'optimisation multi-objectif traditionnelle ont été présentées comment de la notion dominance, les notations diverses pour les espaces de recherche et de solutions réalisables d'un problème, et des classifications pour les algorithmes multi-objectif. Le contenu principal de cette section est présenté dans la suite.

Les problèmes mono-objectif sont caractérisés par l'existence d'un optimum global. En revanche, l'optimisation multi-objectif cherche un ensemble de solutions optimales qui reflètent plusieurs compromis entre tous les critères. Ainsi, l'idée de trouver un optimum unique a été remplacée par la recherche d'un ensemble de éléments qualifiés de solutions qui appartenant à frontière de Pareto, ou simplement, de solutions Pareto optimales. Pour mieux comprendre les concepts d'optimisation multi-objectif et de solution Pareto, on considère d'abord la notation  $\mathbf{X} \subseteq \mathbb{R}^{n_x}$  pour représenter l'espace des variables,  $\mathbf{x}_1$  et  $\mathbf{x}_2$  pour noter deux vecteurs de variables, et un vecteur de fonctions objectif  $\mathbf{f}(\mathbf{x}) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_f}$  pour évaluer les deux vecteurs. Si le décideur préfère  $\mathbf{x}_1$  à  $\mathbf{x}_2$ , il dit que " $\mathbf{x}_1$  domine  $\mathbf{x}_2$ ". Formellement, l'opérateur de dominance  $\preceq$  est défini dans l'espace des objectifs comme :

$$\mathbf{f}(\mathbf{x}_1) \preceq \mathbf{f}(\mathbf{x}_2), \text{ si } \mathbf{f}(\mathbf{x}_1) \leq \mathbf{f}(\mathbf{x}_2), \mathbf{f}(\mathbf{x}_1) \neq \mathbf{f}(\mathbf{x}_2). \quad (2.3)$$

Par analogie, il a été défini aussi l'opérateur  $\succeq$ , et les cas stricts  $\prec$  et  $\succ$ . Maintenant, une fois que l'opérateur de dominance a été défini, le prochaine pas est décrire le problème multi-objectif avec contrainte, donc :

$$\begin{aligned} \min_{\mathbf{x} \in \mathbf{X}} \quad & \mathbf{f}(\mathbf{x}) \\ \text{s.c. : } \quad & \mathbf{g}(\mathbf{x}) \leq \mathbf{0}. \end{aligned} \quad (2.4)$$

L'ensemble solution de ce problème peut être défini comme :

$$\mathbf{X}^* = \{\mathbf{x}^* \in \mathbf{X}' \mid \nexists \mathbf{x} \in \mathbf{X}', \mathbf{f}(\mathbf{x}) \preceq \mathbf{f}(\mathbf{x}^*)\}, \quad (2.6)$$

où  $\mathbf{X}' = \{\mathbf{x} \in \mathbf{X} \mid \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\}$  est l'espace réalisable et  $\mathbf{X}^*$  est l'ensemble de solutions acceptables du problème.

Le choix d'une solution particulière parmi l'ensemble  $\mathbf{X}^*$  dépend du décideur et de la méthode employée. À critère du décideur, il peut utiliser trois sortes de méthodes. Le premier réalise le choix de la solution *à priori*, c'est à dire, ce genre d'algorithme s'agit de mettre des pondérations sur chaque

objectif en transformant un problème multi-objectif en un problème mono-objectif. Il y a des inconvénient en problèmes où la frontière de Pareto est convexe. Les algorithmes qui pertinent au deuxième type méthode, font les choix à *posteriori*, c'est-à-dire, la décision est faite directement sur le  $\mathbf{X}^*$ . Finalement, il y a des algorithmes le choix est fait de façon *progressif*. Dans ce cas, les préférences sont faites pendant le procédé d'optimisation.

*Références : Marler et Arora [80], Andersson [4], Sawaragi et al. [102], Miettinen [84], Yu [127], Osyczka [92], Koski et Silvennoinen [74], Zions [130], Yu [126], Zeleny [128], Waltz [124], Takahashi et al. [119], Benayoun et al. [9], Steuer [116], Messac et al. [83], Messac et al. [82], Das et Dennis [28], Nash Arbitrário, Davis [30], Rao et Freiheit [97], Messac [81].*

## 2.2 - Algorithmes Évolutionnaires

Dans cette section, ils ont été discutés la diversité des Algorithmes Évolutionnaires, cella structure algorithmique, les techniques pour maintenir la diversité dans la frontière de Pareto, et des métriques pour mesurer ces performances.

Les Algorithmes Évolutionnaires sont inspirés par des analogies avec les phénomènes biologiques, par exemple, du processus de sélection naturelle (Algorithme Génétique), du comportement des fourmis (Algorithmes de Colonies de Fourmis) et du déplacement d'un groupe d'oiseaux (Optimisation par Essaims Particulaires). Dans cette thèse, la métaheuristique d'optimisation choisie pour le développement d'algorithme évolutionnaire hybride a été l'Algorithme Génétique. Ainsi, le vocabulaire utilisé pour expliquer cet algorithme a été calqué sur la théorie de l'évolution et de la génétique. Par exemple, le terme *chromosome* est associé à représentation discrète d'une variable ; les *individus* d'une *population* représente les solutions potentielles. Aussi, la sélection, la reproduction, le croisement et la mutation, parmi d'autres analogies, sont employés comme techniques pour traiter les solutions des algorithmes hybrides à la recherche de l'ensemble optimal. La Table 2.3 décrit un exemple d'algorithme génétique multi-objectif.

Les informations principales des dernières sous-sections sont présentées ensuite. La Sous-section 2.2.3 décrit plusieurs méthodes évolutionnaires multi-objectif. La Sous-section 2.2.4 présente les paramètres communément trouvés dans les algorithmes génétiques multi-objectif. La Sous-section 2.2.5 a met en question l'importance de la stratégie élitiste dans le contexte multi-objectif.

La Sous-section 2.2.6 discute cinq métriques pour mesurer les performances et la qualité d'ensemble solution des algorithmes multi-objectif en général. Des métriques présentées sont comparées avec la métrique développée dans cette thèse à la Section 3.7. Plus particulièrement, une métrique trouvée dans la littérature est utilisée dans la phase de validation. Elle s'appelle TE (Taxe d'Erreur) et est basée sur la connaissance préalable d'un ensemble de comparaison, par exemple  $\mathbf{Y}^*$ . La métrique TE est définie par :

$$TE = \frac{\sum_{i=1}^{|\mathbf{A}|} e_i}{|\mathbf{A}|}, \quad (2.12)$$

où  $|\mathbf{A}|$  est le nombre de solutions de l'ensemble  $\mathbf{A}$ , trouvé par la méthode que l'on désire évaluer, et  $e_i$  est tel que :

$$\begin{cases} e_i = 1, & \text{si } e_i \in \mathbf{A} \setminus \mathbf{Y}^* ; \\ e_i = 0, & \text{autrement.} \end{cases} \quad (2.13)$$

La conclusion qui un élément de  $\mathbf{A}$  appartient à  $\mathbf{Y}^*$  est déterminé par une distance Euclidienne, que représente la précision  $\varepsilon_{TE}$  de la métrique TE.

La Sous-section 2.2.7 disserte sur la formation de niche, qui sont techniques que permettent aux algorithmes génétiques de rechercher plusieurs portions d'espace en simultanéité et donc d'éviter que l'algorithme converge à un optimum local.

*Références : Dorigo et Stützle [37], Kennedy et Eberhart [69], Laumanns et al. [76], Deb [32], Knowles [72], Holland [57], Goldberg [45], Soares [109], Gao et al. [44], Schaffer [103], Fosenca et Fleming [42], Murata et Ishibuchi [89], Horn et al. [58], Goldberg et Richardson [46], Srinivas et Deb [115], Deb et al. [34], Coello et Sierra [21], Knowles et Corne [73], Coello et Pulido [20], Knowles [72], Soares et al. [114], Kenneth et Eberhart [69], zitzler [131], Zitzler et al. [132], Veldhuilzen [121], Soares [113], Grefenstette [47], Rump [101], Rudolph [99], Czyzak [24], Hansen et Jaszkiewicz [51], Schott [104], et Zitzler et Thiele [133].*

## 2.3 - Analyse par Intervalles

Tout d'abord, cette section présente des avantages de faire le traitement de données utilisant les intervalles. Les avantages sont la garantie de propagation correcte d'un nombre incertain contenu dans un intervalle. Aussi, des

inconvénients, comme la lenteur des méthodes par intervalles, ont été listés et discutés. Ensuite, un bref historique d'Analyse par Intervalles est présenté.

Les fondements d'Analyse par Intervalles sont basés sur la théorie des ensembles. Donc, la Sous-section 2.3.2 a défini les opérations fondamentales pour traiter les ensembles, et comme conséquence, les intervalles. Les opérations sont présentées par les expressions (2.20-2.26), et des définitions et des propriétés des ensembles sont montrés dans les expressions (2.27-2.30). Les Sous-sections 2.3.3 et 2.3.4 présentent les définitions pour les intervalles réels et pour l'arithmétique pour opérer les intervalles. Pour des raisons pratiques et par simplicité, des définitions ont été réalisées en  $\mathbb{R}$ . Par analogie, des expressions peuvent être étendus pour  $\mathbb{IR}^{n_x}$ , où  $n_x$  représente la dimension du vecteur  $\mathbf{x}$ . Un intervalle réelle  $[x]$  est défini comme :

$$[\underline{x}, \bar{x}] = \{x \in \mathbb{R} \mid \underline{x} \leq x \leq \bar{x}\}, \quad \underline{x} \in \mathbb{R} \text{ et } \bar{x} \in \mathbb{R}, \quad (2.31)$$

et les opérations binaires  $+$ ,  $-$ ,  $*$  et  $/$  comme :

$$[x] \diamond [y] = \{x \diamond y \mid x \in [x], y \in [y]\}. \quad (2.32)$$

Les quatre opérations sont explicitées dans les expressions (2.33-2.38) ; quelques exemples sont présentés ensuite. Dans la Sous-section 2.3.5, les fonctions qui évaluent intervalles sont définis. Formellement, la fonction intervalle  $f([x])$  est définie comme :

$$f([x]) = [\{f(x) \mid x \in [x]\}]. \quad (2.39)$$

Si une variable apparaît plus d'une fois dans une fonction intervalle, alors le résultat de cette fonction est un intervalle plus conservateur, car les incertitudes des intervalles sont propagées. Ce problème est appelé *dépendance*. La Sous-section 2.3.6 explique la dépendance en détails.

Le *vecteur intervalle* ou une *boîte* est décrit dans la Sous-section 2.3.7. Formellement, une boîte  $[\mathbf{x}] \in \mathbb{IR}^{n_x}$  est définie comme :

$$[\underline{\mathbf{x}}, \bar{\mathbf{x}}] = [x_1] \times \dots \times [x_n], \quad (2.42)$$

Normalement, la précision des résultats des méthodes intervalles sont mesurées par *longueur* des intervalles obtenus. La longueur  $w([\mathbf{x}]) : \mathbb{IR}^{n_x} \rightarrow \mathbb{R}$  est définie comme :

$$w([\mathbf{x}]) = \max_{i=\{1, \dots, n_x\}} w([x_i]). \quad (2.43)$$

Aussi, à la fin d'un procédé d'optimisation, le décideur a besoin de prendre une solution réelle dans un ensemble de boîtes. Une estimation très utilisée

est le centre de la boîte. Ainsi, le *centre*  $\mathbf{c}([\mathbf{x}]) : \mathbb{IR}^{n_x} \rightarrow \mathbb{R}^{n_x}$  a été défini par :

$$c_i([\mathbf{x}]) = (\bar{x}_i + \underline{x}_i)/2, \quad i = \{1, \dots, n_x\}. \quad (2.45)$$

Le *volume* de boîtes est une importante mesure dans la technique de niche à être proposée dans le prochaine chapitre. Le volume  $v([\mathbf{x}]) : \mathbb{IR}^{n_x} \rightarrow \mathbb{R}$  est défini par :

$$v([\mathbf{x}]) = \prod_{i=\{1, \dots, n_x\}} w([x_i]). \quad (2.45)$$

Plusieurs méthodes intervalles pour l'optimisation commencent le processus de recherche par une boîte qui contient tout l'espace de projet, et itération par itération, ces méthodes ont besoin d'examiner portions plus petites d'espace pour améliorer la précision. Bissecter d'intervalle se fait nécessaire. La bisection d'une boîte  $[\mathbf{x}]$  en deux autres  $[\mathbf{x}]_1$  et  $[\mathbf{x}]_2$  est défini par :

$$[\mathbf{x}]_1 = [x_1] \times \dots \times [x_j, \frac{x_j + \bar{x}_j}{2}] \times \dots \times [x_n], \quad (2.46)$$

$$[\mathbf{x}]_2 = [x_1] \times \dots \times [\frac{x_j + \bar{x}_j}{2}, \bar{x}_j] \times \dots \times [x_n], \quad (2.47)$$

où  $j = \min\{i \mid w([x_i]) = w([\mathbf{x}])\}$ .

La Sous-section 2.3.8 décrit les *fonctions d'inclusion*, qui sont utilisé par les méthodes intervalles en général. Formellement, considère la fonction  $\mathbf{f} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_f}$ . La fonction  $[\mathbf{f}] : \mathbb{IR}^{n_x} \rightarrow \mathbb{IR}^{n_f}$  est une fonction d'inclusion pour  $\mathbf{f}$  si

$$\forall [\mathbf{x}] \in \mathbb{IR}^{n_x}, \quad \mathbf{f}([\mathbf{x}]) \subseteq [\mathbf{f}]([\mathbf{x}]) \quad (2.49)$$

La Fig.2.4 présente deux exemples de fonctions d'inclusion. Pour une même  $\mathbf{f}$ , il y a plusieurs  $[\mathbf{f}]([\mathbf{x}])$ . Parmi ces fonctions  $[\mathbf{f}]([\mathbf{x}])$ , il est intéressant d'obtenir des fonctions d'inclusion *fine*, *convergentes*, *monotones* et *minimales*. La fonction  $[\mathbf{f}]([\mathbf{x}])$  est fine si :

$$\forall \mathbf{x} \in [\mathbf{x}], \quad \mathbf{f}(\mathbf{x}) = [\mathbf{f}](\mathbf{x}). \quad (2.50)$$

La fonction  $[\mathbf{f}]([\mathbf{x}])$  est convergente si :

$$\lim_{k \rightarrow \infty} w([\mathbf{x}]_k) = 0 \Rightarrow \lim_{k \rightarrow \infty} w([\mathbf{f}]([\mathbf{x}]_k)) = 0. \quad (2.51)$$

La fonction  $[\mathbf{f}]([\mathbf{x}])$  est monotone si :

$$[\mathbf{x}]_k \subset [\mathbf{x}]_{k-1} \Rightarrow [\mathbf{f}]([\mathbf{x}]_k) \subset [\mathbf{f}]([\mathbf{x}]_{k-1}). \quad (2.52)$$

Finalement, la fonction  $[\mathbf{f}]([\mathbf{x}])$  est définie minimale quand elle est la plus petite boîte que contient  $\mathbf{f}(\mathbf{x})$ ,  $\forall \mathbf{x} \in [\mathbf{x}]$ . Les Fig. 2.5 et Fig. 2.6 montrent ces caractéristiques.

Dans les méthodes intervalles, une fonction d'inclusion que trop utilisée est la *fonction d'inclusion naturelle* qui est obtenue en remplaçant la variable  $\mathbf{x}$  par l'intervalle  $[\mathbf{x}]$ , et en remplaçant chaque opérateur d'arithmétique conventionnel ou fonction par son homologue pour les intervalles.

La Sous-section 2.3.9 ont décrit la Logique booléenne pour les intervalles. Dans cette logique, il y a quatre possibilités pour l'ensemble booléen intervalle  $\mathbb{IB}$ , sont-elles :

$$\mathbb{IB} = \{\emptyset, 0, 1, [0, 1]\}. \quad (2.53)$$

où  $\emptyset$  est la absence de connexion entre les ensembles en question, et  $[0, 1]$  signifie le résultat booléen peut être vrai ou faux.

La Sous-section 2.3.10 présente les tests booléens. Ces tests sont importants pour guider les algorithmes dans ses procédés de recherche. Par exemple, les tests booléens peuvent être utilisés pour vérifier si une variable particulière est réalisable ou non via évaluation de fonctions de contraintes, écrites sur la forme de tests d'inclusion. Un exemple de test est présenté dans l'Exemple 8 (page 101).

Un *sous-pavage* de  $\mathbb{R}^n$  est un ensemble de pavés  $\mathbb{R}^n$  ne se recouvrant pas. La Fig. 3.5 montre des sous-pavages que ne se recouvrant pas pour représenter une région selon les contraintes définies en l'Exemple 14 (124). La Sous-section 2.3.11 présente comment les sous-pavages sont organisés en structures de données du type piles et files.

La Sous-section 2.3.12 a cité SIVIA comme une méthode d'inversion ensembliste que s'intéresse à la caractérisation d'espace  $\mathbf{X} \in \mathbb{R}^{n_x}$  donné son image  $\mathbf{Y} \in \mathbb{R}^{n_f}$ . SIVIA utilise plusieurs concepts d'Analyse par Intervalles décrits dans les sous-sections précédents comme bissecter, test d'inclusion et sous-pavages. De nouveau, donnés les fonctions de contraintes dans l'Exemple 14 (124), SIVIA a trouvé l'ensemble des solutions réalisables, que sont montrées en gris dans la figure.

La Sous-section 2.3.13 présente les inéquations intervalles. La notion d'inéquations intervalles est important pour la compréhension du traitement de las fonctions de contraintes.

*Références : Hayes [53], Daumas [29], Moore [85], Jaulin et al. [63], Kearfott et Kreinovich [68], Hansen et Walster [50], Carreras et Walker [15], Morales*

---

et Son [88] en robótica, Soares et al. [111], Siouris et al. [106], Healt [54], Alefeld et Mayer [2], Sunaga [117], Hansen [48], Neumaier [90], Kearfott [67], Hanson [52], Kahan [66], Walster et al. [123], Jaulin et al. [63], Hickey et al. [56], Vaccaro [120], Moore [87], Walster [50], Benhamou et Older [10], et Chabert et Jaulin [17].

## 2.4 - Optimisation Multi-Objectif Intervalle

Cette section montre l'évolution des méthodes qui utilisent l'Analyse par Intervalles dans le contexte d'optimisation, et spécifiquement dans le domaine d'optimisation multi-objectif. Plusieurs travaux de recherche en cette direction sont cités. Après la partie historique, cette section a discuté des points de vue *optimiste* et *pessimiste* pour la comparaison entre intervalles. La Fig. 2.8 a présenté la comparaison de dominance des intervalles ; le graphique centrale a présenté le résultat en utilisant un point de vue optimiste et le graphique à droite un point de vue pessimiste. Dans les algorithmes proposés, le pessimisme a été préféré comme stratégie à être utilisé durant les comparaisons de dominance, car il maintient des boîtes qui peuvent contenir des solutions pour le problème en question.

*Références : Moore [85], Skelboe [107], Moore [86], Alefeld [1], Hansen [49], Skelboe [108], Ichida et Fujii [60], Hansen et Walster [50], Ichida et Fujii [61], Mahato et Bhunia [78], Casado et al. CasadoGarcíaMartínezSergeyev2003, Csallener et al.[23], Sengupta et Pal [105], Ruetsch [100], Barichard et Hao [6], Benhamou et Older [10], et Jaulin [62].*

---

## Chapitre 3 - Algorithmes Intervalles Déterministes et Évolutionnaire pour l'Optimisation Robuste Multi-Objectif

Au début du chapitre, pour introduire les algorithmes intervalles, quelques définitions de robustesse d'autres domaines de recherche ont été présentées. Par exemple, la notion de robustesse en Ingénierie du Logiciel a été associée à tolérance aux erreurs, lorsqu'en biologie, la robustesse c'est une caractéristique d'un individu de maintenir ses fonctionnalités même quand l'ambiant change. Dans le contexte d'optimisation, la robustesse d'une solution peut être interprétée comme la capacité de maintenir sa réponse au système, même s'il y a des perturbations. Cependant, on peut aussi définir la robustesse d'une solution en regardant seulement sa performance du *scénario du pire cas*. Dans la littérature, l'optimisation robuste a été décrite de deux façons : *stochastique* et *déterministe*. Les méthodes basées sur probabilités utilisant techniques d'échantillonnage pour mesurer la robustesse d'une solution. Donc, la qualité du résultat dépend de la qualité d'échantillonnage. D'autre part, les méthodes déterministes calculent la robustesse d'une solution considérant toutes les interférences des incertitudes. Dans plusieurs études, les intervalles ont été utilisés pour réaliser la tâche de traitement des incertitudes. C'est donc l'optimisation robuste envisagée par ce travail : traiter le problème robuste multi-objectif au moyen des intervalles pour calculer le scénario du pire cas.

*Références : Kitano [71], De Vale et Koopman [35], Du et Chen [38], Parkinson et al. [93], Kouvelis et Yu [75], Roy [98], Dias et Clímaco [36], Beyer et Sendhoff [12], Jin et Branke [65], Deb et Gupta [33], et Soares et al. [110].*

### 3.1 - Bref Historique de l' Optimisation Robuste

Cette section a exposé brièvement quelques travaux généralement cités comme importants pour la démarche de l'optimisation robuste. Les premières études qu'ont pris en compte des incertitudes apparaissaient dans les années 50 en travaux de statistique pour améliorer la production agricole, par l'utilisation de la programmation linéaire sous incertitudes avec contraintes, et pour améliorer la qualité en ingénierie dans le cadre du développement de produits dès la phase de conception. Dès ce début, l'optimisation robuste a évolué en plusieurs directions, par exemple, pour traiter les problèmes convexes, les systèmes discrets et les systèmes dynamiques.

*Références : Fisher [41], Dantzig [26], Charnes et Cooper [18], Taguchi [118], Ben-Tal et Nemirovski [7], Ben-Tal et Nemirovski [8], El Ghaoui et al. [40], El Ghaoui et Calafiori [39], Kouvelis et Yu [75], Bertsimas et Sim [11], Mareschal [79], et Dias et Clímaco [36].*

### 3.2 - Concepts et Notation

Les relations de dominance sont utilisées par les méthodologies multi-objectif pour réaliser les tâches de classification, de sélection et d'élimination de solutions dans des procédés de recherche. Donc, il est important de définir des opérateurs pour établir la supériorité ou l'infériorité d'une solution ou d'autre. Une fois que les algorithmes de cette thèse utilisent des données réelles et intervalles simultanément, il est nécessaire redéfinir l'opérateur  $\preceq$  comme montré ci dessous.

**Définition 16** Soient  $\mathbf{u} \in \mathbb{R}^{n_n}$ ,  $\mathbf{v} \in \mathbb{R}^{n_n}$ ,  $[\mathbf{t}] \in \mathbb{IR}^{n_n}$  et  $[\mathbf{z}] \in \mathbb{IR}^{n_n}$ . L'opérateur de comparaison  $\preceq$  est redéfini comme :

$$\mathbf{u} \preceq \mathbf{v} \Leftrightarrow \mathbf{u} \leq \mathbf{v}, \mathbf{u} \neq \mathbf{v}; \quad (3.1)$$

$$\mathbf{u} \preceq [\mathbf{t}] \Leftrightarrow \mathbf{u} \preceq \mathbf{t}; \quad (3.2)$$

$$[\mathbf{z}] \preceq [\mathbf{t}] \Leftrightarrow \bar{\mathbf{z}} \preceq \underline{\mathbf{t}}. \quad (3.3)$$

Utilisant la notion de dominance, il est possible de distinguer, dans un ensemble de vecteurs : un groupe composé pour les vecteurs plus petits, autre composé pour les vecteurs plus grands et un vecteur qui représente le point idéal de maximisation. La définition de chacun est décrite ensuite.

**Définition 17** Considérons l'ensemble de vecteurs  $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m\}$ ,  $\mathbf{U} \subseteq \mathbb{R}^{n_n}$ . Les plus petits vecteurs de  $\mathbf{U}$  définissent l'ensemble  $\Upsilon_{\preceq}$  décrit comment :

$$\Upsilon_{\preceq}(\mathbf{U}) = \{\mathbf{u} \in \mathbf{U} \mid \nexists \mathbf{u}' \in \mathbf{U}, \mathbf{u}' \preceq \mathbf{u}\}. \quad (3.5)$$

Par analogie, les plus grands vecteurs de  $\mathbf{U}$  composent l'ensemble  $\Upsilon_{\succeq}$  décrit comme :

$$\Upsilon_{\succeq}(\mathbf{U}) = \{\mathbf{u} \in \mathbf{U} \mid \nexists \mathbf{u}' \in \mathbf{U}, \mathbf{u} \preceq \mathbf{u}'\}. \quad (3.6)$$

**Définition 18** Considérons l'ensemble de vecteurs  $\mathbf{U} = \{\mathbf{x}, \mathbf{y}, \dots, \mathbf{z}\} \subset \mathbb{R}^{n_n}$ . Le point idéal de maximisation  $\mathbf{u}^{\max} \in \mathbb{R}^{n_n}$  de l'ensemble  $\mathbf{U}$  est :

$$u_i^{\max}(\mathbf{U}) = \max_i \{x_i, y_i, \dots, z_i\}, \quad i = 1, \dots, n_n. \quad (3.7)$$

Pour faire suite à cette analyse, un théorème a été formulé pour présenter la relation entre  $\mathbf{u}^{\max}$  et  $\Upsilon_{\succeq}$ . Ce théorème amène des conséquences importantes pour l'optimisation robuste décrite sur les prochaines sections.

**Théorème 1** Considère  $\mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^{n_x}$ ,  $\mathbf{f}(\mathbf{x}) : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_f}$  et  $\mathbf{U} = \{\mathbf{u} \in \mathbb{R}^{n_f} \mid \forall \mathbf{x} \in \mathbf{X}, \mathbf{u} = \mathbf{f}(\mathbf{x})\}$ ,  $\mathbf{U} \neq \emptyset$ . Donc,

$$\Upsilon_{\succeq}(\mathbf{U}) \neq \{\mathbf{u}^{\max}(\mathbf{U})\} \Rightarrow \mathbf{u}^{\max}(\mathbf{U}) \notin \mathbf{U}. \quad \blacksquare \quad (3.8)$$

**Preuve :** Assume  $\mathbf{u}^{\max}(\mathbf{U}) \in \Upsilon_{\succeq}(\mathbf{U})$ . Utilisant (3.6) et (3.7) on a conclu que  $\Upsilon_{\succeq}(\mathbf{U})$  est l'ensemble unitaire  $\Upsilon_{\succeq}(\mathbf{U}) = \{\mathbf{u}^{\max}(\mathbf{U})\}$ . Par définition, en utilisant seulement (3.7) on a conclu que  $\forall \mathbf{u} \in \mathbf{U}$ ,  $\mathbf{u}^{\max}(\mathbf{U}) \succeq \mathbf{u}$ . Donc, si  $\mathbf{u}^{\max}(\mathbf{U}) \notin \Upsilon_{\succeq}(\mathbf{U})$ , alors  $\mathbf{u}^{\max}(\mathbf{U}) \notin \mathbf{U}$ .

### 3.3 - Le Problème Multi-Objectif Robuste

Comme il a été précisé dans l'introduction de ce chapitre, la notion de robustesse enveloppe différents concepts. Pour ce travail, il a été choisi la philosophie de la robustesse afin de prendre en compte le pire cas de chaque solution évalué. Donc, le problème d'optimisation robuste considéré peut être défini comme présenté ensuite.

Considérons le vecteur de variables de projet  $\mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^{n_x}$  et le vecteur de paramètres d'incertitude  $\mathbf{p} \in \mathbf{P} \subseteq \mathbb{R}^{n_p}$ . En définissant le vecteur de fonctions objectif par  $\mathbf{f}(\mathbf{x}, \mathbf{p}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_f}$  et le vecteur de contrainte par  $\mathbf{g}(\mathbf{x}, \mathbf{p}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_g}$ , le problème de minimisation robuste multi-objectif peut être écrit comment :

$$\begin{aligned} \min_{\mathbf{x} \in \mathbf{X}} \quad & \max_{\mathbf{p} \in \mathbf{P}} \mathbf{f}(\mathbf{x}, \mathbf{p}) \\ \text{s.c.} \quad & \mathbf{g}(\mathbf{x}, \mathbf{p}) \leq \mathbf{0}. \end{aligned} \quad (3.9)$$

Définissons l'espace réalisable par :

$$\mathbf{X}' = \{\mathbf{x} \in \mathbf{X} \mid \forall \mathbf{p} \in \mathbf{P}, \mathbf{g}(\mathbf{x}, \mathbf{p}) \leq \mathbf{0}\}. \quad (3.10)$$

Résoudre (3.9) consiste en trouver l'ensemble de minimiseurs robustes  $\mathbf{X}^* \subseteq \mathbf{X}'$  pour le pire cas d'influence des incertitudes. Formellement,

$$\mathbf{X}^* = \{\mathbf{x}^* \in \mathbf{X}' \mid \nexists \mathbf{x} \in \mathbf{X}', \max_{\mathbf{p} \in \mathbf{P}} \mathbf{f}(\mathbf{x}, \mathbf{p}) \preceq \max_{\mathbf{p}' \in \mathbf{P}} \mathbf{f}(\mathbf{x}^*, \mathbf{p}')\}. \quad (3.11)$$

L'espace des objectifs est aussi objet de cette étude. Ainsi, considère les images des fonctions  $\mathbf{y} = \mathbf{f}(\mathbf{x}, \mathbf{p})$  et  $\mathbf{Y}' = \mathbf{f}(\mathbf{X}', \mathbf{P})$ , où :

$$\mathbf{f}(\mathbf{X}', \mathbf{P}) = \bigcup_{\mathbf{x} \in \mathbf{X}', \mathbf{p} \in \mathbf{P}} \mathbf{f}(\mathbf{x}, \mathbf{p}). \quad (3.12)$$

Comme conséquent du Théorème 1, si pour quelque solution  $\mathbf{x}^*$  le processus de maximisation en (3.9) produit  $\mathbf{u}^{\max}(\mathbf{f}(\mathbf{x}^*, \mathbf{p})) \notin \Upsilon_{\succeq}(\mathbf{f}(\mathbf{x}^*, \mathbf{p}))$ , il est possible que  $\mathbf{u}^{\max}(\mathbf{f}(\mathbf{X}^*, \mathbf{P})) \notin \mathbf{Y}'$ . Ainsi, pour représenter l'image robuste de toutes les solutions  $\mathbf{X}^*$  du problème (3.9) est nécessaire définir un espace que contienne  $\mathbf{u}^{\max}(\mathbf{f}(\mathbf{x}^*, \mathbf{p}))$ ,  $\forall \mathbf{x}^* \in \mathbf{X}^*$  et  $\forall \mathbf{p} \in \mathbf{P}$ . Ceci est fait en replaçant l'image  $\mathbf{f}(\mathbf{X}', \mathbf{P})$  par la boîte  $[\mathbf{Y}]'$  comme :

$$\overline{[Y_i]}' = \max_{\mathbf{p} \in \mathbf{P}} f_i(\mathbf{x}, \mathbf{p}), \quad \forall \mathbf{x} \in \mathbf{X}', i = \{1, \dots, n_f\}; \quad (3.13)$$

$$\underline{[Y_i]}' = \min_{\mathbf{p} \in \mathbf{P}} f_i(\mathbf{x}, \mathbf{p}), \quad \forall \mathbf{x} \in \mathbf{X}', i = \{1, \dots, n_f\}. \quad (3.14)$$

Maintenant, si  $\mathbf{Y}^*$  représente la *frontière de Pareto robuste* ;  $\mathbf{Y}^+$  et  $\mathbf{Y}^-$  les *portions au-dessus* et *au-dessous de la frontière robuste*, respectivement. Les images  $\mathbf{Y}^*$ ,  $\mathbf{Y}^+$  et  $\mathbf{Y}^-$  sont liées par la relation :

$$[\mathbf{Y}]' = \mathbf{Y}^- \cup \mathbf{Y}^+, \quad \mathbf{Y}^* \subseteq \mathbf{Y}^+, \quad (3.15)$$

avec :

$$\mathbf{Y}^+ = \{\mathbf{y}^+ \in [\mathbf{Y}]' \mid \exists \mathbf{x} \in \mathbf{X}, \forall \mathbf{p} \in \mathbf{P}, \mathbf{f}(\mathbf{x}, \mathbf{p}) \preceq \mathbf{y}^+\}; \quad (3.16)$$

$$\mathbf{Y}^- = \{\mathbf{y}^- \in [\mathbf{Y}]' \mid \mathbf{y} \in [\mathbf{Y}]', \mathbf{y} \notin \mathbf{Y}^+\}; \quad (3.17)$$

$$\mathbf{Y}^* = \{\mathbf{y}^* \in [\mathbf{Y}]' \mid \nexists \mathbf{y} \in \mathbf{Y}^+, \mathbf{y} \preceq \mathbf{y}^*\}. \quad (3.18)$$

La Fig. 3.1 (page 115) présente un problème robuste de façon générale: le côté à droit illustre les régions  $\mathbf{Y}^*$ ,  $\mathbf{Y}^+$  et  $\mathbf{Y}^-$  ; le côté à gauche montre l'espace de recherche et l'espace des incertitudes. Pour illustrer le problème robuste, cette section termine avec deux exemples simples.

### 3.4 - Algorithme Multi-Objectif Robuste par Intervalles I

Le premier algorithme pour traiter le problème (3.9) était appelé de [I]RMOA I (Algorithme Multi-Objectif Robuste par Intervalles I). En fait, cet algorithme ne cherche pas trouver  $\mathbf{Y}^*$ , mais construit une enveloppe discrète sur  $\mathbf{Y}^*$ . Ainsi, le [I]RMOA I a pour la finalité de valider d'autres méthodes ; une méthode est considérée valide si l'image de son ensemble trouvé  $\mathbf{f}(\mathbf{X}^*)$  appartient à l'enveloppe sur  $\mathbf{Y}^*$ , calculé par [I]RMOA I. Par la suite, il est présenté comme l'enveloppe était obtenue.

**Définition 19** Considérons  $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$ ,  $\mathbf{U} \subset \mathbf{Y}^-$ . La représentation discrète de  $\mathbf{Y}^-$  lié à  $\mathbf{U}$  est :

$$\mathbf{K}^- = \{\mathbf{u}' \in [\mathbf{Y}]' \mid \exists \mathbf{u} \in \Upsilon_{\succeq}(\mathbf{U}), \mathbf{u}' \preceq \mathbf{u}\}. \quad (3.21)$$

Par analogie, si  $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m\}$ ,  $\mathbf{V} \subset \mathbf{Y}^+$ , donc la représentation discrète de  $\mathbf{Y}^+$  lié à  $\mathbf{V}$  est :

$$\mathbf{K}^+ = \{\mathbf{v}' \in [\mathbf{Y}]' \mid \exists \mathbf{v} \in \Upsilon_{\preceq}(\mathbf{V}), \mathbf{v}' \succeq \mathbf{v}\}. \quad (3.22)$$

**Définition 20** La frontière supérieure à  $\mathbf{K}^-$  et la frontière inférieure à  $\mathbf{K}^+$  sont définies comme :

$$\partial\mathbf{K}^- = \{\mathbf{u} \in \mathbf{K}^- \mid \nexists \mathbf{u}' \in \mathbf{K}^-, \mathbf{u} \preceq \mathbf{u}'\}; \quad (3.23)$$

$$\partial\mathbf{K}^+ = \{\mathbf{v} \in \mathbf{K}^+ \mid \nexists \mathbf{v}' \in \mathbf{K}^+, \mathbf{v}' \preceq \mathbf{v}\}. \quad (3.24)$$

**Définition 21** Symbolisant  $\setminus$  comment l'opérateur complément, les portions discrètes non obtenues directement de  $\mathbf{Y}^-$  et de  $\mathbf{Y}^+$  sont définies par :

$$\Delta\mathbf{K}^- = (\mathbf{Y}^- \setminus \mathbf{K}^-) \cup \partial\mathbf{K}^-; \quad (3.25)$$

$$\Delta\mathbf{K}^+ = (\mathbf{Y}^+ \setminus \mathbf{K}^+) \cup \partial\mathbf{K}^+. \quad (3.26)$$

**Théorème 2** Considérant la précision de  $\mathbf{K}^-$  et  $\mathbf{K}^+$ , on a l'inclusion suivante :

$$\mathbf{Y}^* \subset \Delta\mathbf{K}^- \cup \Delta\mathbf{K}^+. \quad \blacksquare$$

**Preuve:** De (3.15),  $[\mathbf{Y}]' = \mathbf{Y}^- \cup \mathbf{Y}^+$ . De (3.25) et (3.26),  $\mathbf{Y}^- = \mathbf{K}^- \cup \Delta\mathbf{K}^-$ ,  $\mathbf{K}^- \cap \Delta\mathbf{K}^- = \partial\mathbf{K}^-$  et  $\mathbf{Y}^+ = \mathbf{K}^+ \cup \Delta\mathbf{K}^+$ ,  $\mathbf{K}^+ \cap \Delta\mathbf{K}^+ = \partial\mathbf{K}^+$ . En remplaçant  $\mathbf{Y}^-$  et  $\mathbf{Y}^+$  en (3.15), il vient :

$$[\mathbf{Y}]' = (\mathbf{K}^- \cup \Delta\mathbf{K}^-) \cup (\mathbf{K}^+ \cup \Delta\mathbf{K}^+).$$

Mais,  $\mathbf{K}^- \cap \Delta\mathbf{K}^- = \partial\mathbf{K}^-$  et  $\mathbf{K}^+ \cap \Delta\mathbf{K}^+ = \partial\mathbf{K}^+$  et ainsi, il existe des occurrences extras de  $\partial\mathbf{K}^-$  et  $\partial\mathbf{K}^+$  qui peuvent être éliminées. Donc,

$$\begin{aligned} [\mathbf{Y}]' &= ((\mathbf{K}^- \setminus \partial\mathbf{K}^-) \cup \Delta\mathbf{K}^-) \cup ((\mathbf{K}^+ \setminus \partial\mathbf{K}^+) \cup \Delta\mathbf{K}^+); \\ [\mathbf{Y}]' &= ((\mathbf{K}^- \setminus \partial\mathbf{K}^-) \cup (\mathbf{K}^+ \setminus \partial\mathbf{K}^+)) \cup (\Delta\mathbf{K}^- \cup \Delta\mathbf{K}^+). \end{aligned}$$

Les expressions (3.18), (3.21) et (3.22) assurent que  $\mathbf{Y}^* \not\subseteq (\mathbf{K}^- \setminus \partial\mathbf{K}^-)$  et  $\mathbf{Y}^* \not\subseteq (\mathbf{K}^+ \setminus \partial\mathbf{K}^+)$ . Donc,

$$\mathbf{Y}^* \subset \Delta\mathbf{K}, \text{ soient } \Delta\mathbf{K} = \Delta\mathbf{K}^- \cup \Delta\mathbf{K}^+. \quad \blacksquare$$

En fait, l'Équation (3.18) garantie que  $\mathbf{Y}^* \subset \Delta\mathbf{K}^+$ . Mais  $\Delta\mathbf{K}^+$  seul ne peut pas directement être enveloppée. La Fig. 3.3 illustre la représentation des approximations robustes.

Maintenant, nous présentons la méthode [I]RMOA I. L'algorithme commence à partir d'ensemble d'échantillons (solutions possibles) uniformément distribués dans l'espace de recherche et de façon aléatoire dans l'espace des incertitudes. À la suite, ces échantillons sont évalués par le vecteur des fonctions objectif  $\mathbf{f}(\mathbf{x}, \mathbf{p})$ , et les résultats sont gardés dans la file  $\mathbf{Q}_{\text{fxp}}$ . En fait, l'échantillonnage peut se donner directement dans l'espace des objectifs, mais il y a aussi des inconvénients. Chaque élément de  $\mathbf{Q}_{\text{fxp}}$  est testé s'il appartient à  $\mathbf{K}^-$  ou à  $\mathbf{K}^+$ . Après la classification de tous les éléments, [I]RMOA I a l'information suffisante pour trouver  $\Delta\mathbf{K}$ . Par exemple,  $\Delta\mathbf{K} = [\mathbf{Y}]' \setminus (\mathbf{K}^+ \cup \mathbf{K}^-)$ , et comme conséquence  $\Delta\mathbf{K} \supset \mathbf{Y}^*$ .

L'algorithme [I]RMOA I est présenté dans la Table 3.1, et ses paramètres sont décrits dans la Table 3.2. La description schématique des algorithmes FPS et CSC est donnée seulement dans le texte principal du mémoire à la Table 3.3 et à la Table 3.4 respectivement. Le plus important c'est le savoir que les deux algorithmes FPS et CSC retournent une seule conclusion parmi des options: a) il existe au moins un minimiseur  $\mathbf{x}^*$  tel que  $\mathbf{f}(\mathbf{x}^*, \mathbf{p}) \preceq \mathbf{y}$ ; b) il n'existe pas un minimiseur  $\mathbf{x}^*$  tel que  $\mathbf{f}(\mathbf{x}^*, \mathbf{p}) \preceq \mathbf{y}$ . Si FPS et CSC retournent (a), alors  $\mathbf{y}$  appartient à  $\mathbf{Q}_{\mathbf{K}^+}$ ; autrement,  $\mathbf{y}$  appartient à  $\mathbf{Q}_{\mathbf{K}^-}$ .

*Références: Soares et al. [112] et Jaulin et al. [64].*

Table 3.1: Algorithme [I]RMOA I.

entrée	sortie
$([\mathbf{X}], [\mathbf{P}], \varepsilon_{\mathbf{x}\#}, \varepsilon_{[\mathbf{x}]})$	$(\mathbf{K}^-, \mathbf{K}^+)$
<ol style="list-style-type: none"> <li>1 Vider les files <math>\mathbf{Q}_{\mathbf{fxp}}</math>, <math>\mathbf{Q}_{\mathbf{K}^+}</math> et <math>\mathbf{Q}_{\mathbf{K}^-}</math>.</li> <li>2 Appeler la fonction <math>\text{grid}([\mathbf{X}], [\mathbf{P}], \varepsilon_{\mathbf{x}\#})</math> pour générer un ensemble de vecteurs <math>\mathbf{x}</math>, uniformément distribué dans <math>[\mathbf{X}]</math>. Pour chaque <math>\mathbf{x}</math>, un vecteur <math>\mathbf{p}</math> est choisi par événement aléatoire dans <math>[\mathbf{P}]</math>, et évaluer le pair <math>(\mathbf{x}, \mathbf{p})</math> avec <math>\mathbf{f}(\mathbf{x}, \mathbf{p})</math>, enfiler le résultat à <math>\mathbf{Q}_{\mathbf{fxp}}</math>.</li> <li>3 Si <math>\mathbf{Q}_{\mathbf{fxp}}</math> est vide, alors aller au <i>Pas 7</i>, sinon exécute: <ol style="list-style-type: none"> <li>3.a défiler <math>\mathbf{y}</math> de <math>\mathbf{Q}_{\mathbf{fxp}}</math> ;</li> <li>3.b vérifier si il existe un vecteur <math>\mathbf{v} \in \mathbf{Q}_{\mathbf{K}^+}</math>, tel que <math>\mathbf{v} \preceq \mathbf{y}</math> et si il existe un vecteur <math>\mathbf{u} \in \mathbf{Q}_{\mathbf{K}^-}</math>, tel que <math>\mathbf{y} \preceq \mathbf{u}</math>. Ssi ces conditions seraient fausses, aller au <i>Pas 4</i>. Autrement, aller au <i>Pas 3</i>.</li> </ol> </li> <li>4 Appeler les fonctions <math>\text{FPS}([\mathbf{P}], [\mathbf{X}], \mathbf{y}, \varepsilon_{[\mathbf{x}]})</math> et <math>\text{CSC}([\mathbf{x}], \mathbf{y}, [\mathbf{P}], \varepsilon_{[\mathbf{x}]})</math> par vérifier si <math>\mathbf{y}</math> appartient à <math>\mathbf{K}^+</math> ou si <math>\mathbf{y}</math> appartient à <math>\mathbf{K}^-</math>.</li> <li>5 Si <math>\mathbf{y}</math> appartient à <math>\mathbf{K}^+</math>, alors supprimer en <math>\mathbf{Q}_{\mathbf{K}^+}</math> les éléments <math>\mathbf{v}</math> tels que <math>\mathbf{y} \preceq \mathbf{v}</math> et enfiler <math>\mathbf{y}</math> à <math>\mathbf{Q}_{\mathbf{K}^+}</math>. Autrement, supprimer dans <math>\mathbf{Q}_{\mathbf{K}^-}</math> les éléments <math>\mathbf{u}</math> tels que <math>\mathbf{u} \preceq \mathbf{y}</math> et enfiler <math>\mathbf{y}</math> à <math>\mathbf{Q}_{\mathbf{K}^-}</math>.</li> <li>6 Aller au <i>Pas 3</i>.</li> <li>7 Calculer <math>\mathbf{K}^-</math>, utilisant (3.21), par chaque <math>\mathbf{u} \in \mathbf{Q}_{\mathbf{K}^-}</math>.</li> <li>8 Calculer <math>\mathbf{K}^+</math>, utilisant (3.22), par chaque <math>\mathbf{v} \in \mathbf{Q}_{\mathbf{K}^+}</math>.</li> <li>9 Retourner <math>\mathbf{K}^-</math> et <math>\mathbf{K}^+</math>.</li> </ol>	

Table 3.2: Symboles utilisé par [I]RMOA I.

Symboles	Description	Type
$[\mathbf{X}]$	espace de recherche initial	$\mathbb{IR}^{n_x}$
$[\mathbf{P}]$	espace d'incertitude initial	$\mathbb{IR}^{n_p}$
$\mathbf{f}(\mathbf{x}, \mathbf{p})$	fonction objectif robuste	$\mathbb{R}^{n_f}$
$\mathbf{y}$	résultat de $\mathbf{f}(\mathbf{x}, \mathbf{p})$	$\mathbb{R}^{n_f}$
$\varepsilon_{\mathbf{x}\#}$	précision requise pour le grid	$\mathbb{R}$
$\varepsilon_{[\mathbf{x}]}$	précision requise pour le $[\mathbf{x}]$	$\mathbb{R}$
$\mathbf{Q}_{\mathbf{fxp}}$	échantillons évalués par $\mathbf{f}(\mathbf{x}, \mathbf{p})$	File: $\mathbb{R}^{n_f}$
$\mathbf{Q}_{[\mathbf{x}]}$	sous-pavages de $[\mathbf{X}]$	File: $\mathbb{IR}^{n_x}$
$\mathbf{Q}_{[\mathbf{p}]}$	sous-pavages de $[\mathbf{P}]$	File: $\mathbb{IR}^{n_p}$
$\mathbf{Q}_{\mathbf{K}^+}$	vecteurs que définissent $\mathbf{K}^+$	File: $\mathbb{R}^{n_f}$
$\mathbf{Q}_{\mathbf{K}^-}$	vecteurs que définissent $\mathbf{K}^-$	File: $\mathbb{R}^{n_f}$

### 3.5 - Algorithme Multi-Objectif Robuste par Intervalles II

Différemment du [I]RMOA I, la méthode [I]RMOA II (Algorithme Multi-Objectif Robuste par Intervalles II) cherche envelopper  $\mathbf{X}^*$ , alors que l'autre a pour l'objectif d'envelopper  $\mathbf{Y}^*$ . Spécifiquement, le but principal de l'algorithme [I]RMOA II est de trouver tous les sous-pavages  $[\mathbf{x}]$  de l'espace de recherche initial  $[\mathbf{X}]$ , avec une précision  $\varepsilon_{[\mathbf{x}]}$ , tels que le pire cas de ses images est non dominé entre se. Les images sont obtenues en prenant en compte l'action des incertitudes sur chaque  $[\mathbf{x}]$ . Aussi, il est possible estimer les valeurs réelles des éléments de  $\mathbf{X}^*$  en prenant le centre de chaque boîte solution, notée par  $\mathbf{c}([\mathbf{x}]^*)$ . Avant de décrire le processus de recherche du [I]RMOA II, il est nécessaire présenter quelques notions fondamentales au développement des algorithmes, comment: a) la représentation de l'espace d'incertitude  $[\mathbf{P}]$  par sous-pavages ; b) le traitement de contraintes ; c) la procédure de calculer la fonction d'inclusion et le pire cas ; et d) la élimination de région non robuste.

L'espace des incertitudes  $[\mathbf{P}]$  est remplacé par sous-pavages  $[\mathbf{p}]$  qui ne se recouvrent pas généré d'un nombre fini de bisections de  $[\mathbf{P}]$ . Formellement,

$$[\mathbf{P}] = \bigcup_{i=1, \dots, n_{sp}} [\mathbf{p}]_i. \quad (3.27)$$

soient  $n_{sp}$  le nombre de sous-pavages généré, en considérant une précision  $\varepsilon_{[\mathbf{p}]}$ . Ces sous-pavages sont enregistrés dans une file  $\mathbf{Q}_{[\mathbf{p}]}$ . La Table 3.5 présente l'algorithme de calcul de sous-pavages.

Table 3.5: Décomposition en sous-pavages.

	entrée ( $[\mathbf{P}], \varepsilon_{[\mathbf{p}]}$ )	sortie ( $\mathbf{Q}_{[\mathbf{p}]}$ )
1	initialiser $\mathbf{Q}_{[\mathbf{p}]}$ avec $[\mathbf{P}]$	
2	si $w(\mathbf{Q}_{[\mathbf{p}]\{1\}}) \leq \varepsilon_{[\mathbf{p}]}$ , alors aller au <i>Pas 5</i>	
3	défiler $[\mathbf{p}]$ de $\mathbf{Q}_{[\mathbf{p}]}$	
4	bissecter $[\mathbf{p}]$ , enfiler les résultats à $\mathbf{Q}_{[\mathbf{p}]}$ et aller au <i>Pas 2</i>	
5	retourner $\mathbf{Q}_{[\mathbf{p}]}$	

Le traitement de contraintes a été fait en utilisant des tests d'inclusion. En considérant que les fonctions de contraintes écrites comme en (3.9), les tests d'inclusion pour ces cas peuvent être formulés comme :

$$[t]([\mathbf{x}]) = \begin{cases} 1, & \text{si } [\mathbf{g}]([\mathbf{x}], [\mathbf{p}]) \subseteq [-\infty, 0], \\ 0, & \text{si } [\mathbf{g}]([\mathbf{x}], [\mathbf{p}]) \subset [0, \infty], \text{ et} \\ [0, 1], & \text{autrement.} \end{cases} \quad (3.30)$$

L'algorithme pour traiter les contraintes est présenté dans la Table 3.6.

Table 3.6: Algorithme pour traiter les contraintes.

	entrée $([\mathbf{X}], [\mathbf{P}], \varepsilon_{[\mathbf{x}]}, \varepsilon_{[\mathbf{p}]})$	sortie $(\mathbf{Q}'_{[\mathbf{x}]}, \mathbf{Q}_{[\mathbf{p}]})$
1	initialiser $\mathbf{Q}'_{[\mathbf{x}]}$ avec $\emptyset$ , $\mathbf{Q}_{[\mathbf{x}]}$ avec $[\mathbf{X}]$ et $\mathbf{Q}_{[\mathbf{p}]}$ avec $[\mathbf{P}]$	
2	gérer sous-pavages(entrée: $[\mathbf{P}], \varepsilon_{[\mathbf{p}]}$ ; sortie: $\mathbf{Q}_{[\mathbf{p}]}$ ) (Table 3.5)	
3	si $\mathbf{Q}_{[\mathbf{x}]}$ est vide, alors aller au <i>Pas 8</i>	
4	défiler $[\mathbf{x}]$ de $\mathbf{Q}_{[\mathbf{x}]}$	
5	si $[\mathbf{g}]([\mathbf{x}], \mathbf{Q}_{[\mathbf{p}]}) \subset [0, \infty]$ alors aller au <i>Pas 3</i>	
6	si $[\mathbf{g}]([\mathbf{x}], \mathbf{Q}_{[\mathbf{p}]}) \subseteq [-\infty, 0]$ alors enfiler $[\mathbf{x}]$ à $\mathbf{Q}'_{[\mathbf{x}]}$ , aller au <i>Pas 3</i>	
7	si $w([\mathbf{x}]) \geq \varepsilon_{[\mathbf{x}]}$ , bissecter $[\mathbf{x}]$ , enfiler les résultats à $\mathbf{Q}_{[\mathbf{x}]}$ et aller au <i>Pas 3</i>	
8	retourner $\mathbf{Q}'_{[\mathbf{x}]}$ et $\mathbf{Q}_{[\mathbf{p}]}$	

L'Exemple 14 traite un cas avec cinq contraintes. Son résultat est présenté dans la Fig.3.5 (page 125). En prenant compte cette figure, la région réalisable ( $[t]([\mathbf{x}]) = 1$ ) est en gris clair, la région non réalisable ( $[t]([\mathbf{x}]) = 0$ ) est en blanc, et le gris foncé exhibe une région ambiguë ( $[t]([\mathbf{x}]) = [0, 1]$ ).

Dans cette thèse toutes les fonctions d'inclusion utilisées sont du type naturel. Les calculs des fonctions objectif robustes d'inclusion ont été faites pour une boîte  $[\mathbf{x}]$  qui représente une région, et pour un vecteur  $\mathbf{x}$  qui représente une possible solution dedans  $[\mathbf{x}]$ . Donc, en considérant  $\mathbf{x}$  la fonction d'inclusion est notée par  $[\mathbf{f}](\mathbf{x}, [\mathbf{p}])$ ,  $[\mathbf{p}] \in \mathbf{Q}_{[\mathbf{p}]}$ . Ainsi, le pire cas de  $[\mathbf{f}](\mathbf{x}, [\mathbf{p}])$ , est le point idéal de maximisation  $\overline{[\mathbf{f}]}_{\mathbf{x}}^*$  calculé par :

$$\overline{[f_i]}_{\mathbf{x}}^* = \max_{[\mathbf{p}] \in \mathbf{Q}_{[\mathbf{p}]}} \overline{[f_i]}(\mathbf{x}, [\mathbf{p}]), \quad i = \{1, \dots, n_f\}; \quad (3.28)$$

$$\underline{[f_i]}_{\mathbf{x}}^* = \min_{[\mathbf{p}] \in \mathbf{Q}_{[\mathbf{p}]}} \underline{[f_i]}(\mathbf{x}, [\mathbf{p}]), \quad i = \{1, \dots, n_f\}. \quad (3.29)$$

Dans [I]RMOA II, la *Fonction d'Inclusion Minimale*  $\text{MIF}(\mathbf{x}, \mathbf{Q}_{[\mathbf{p}]}) = [\mathbf{f}]_{\mathbf{x}}^*$ . Par analogie,  $\text{MIF}([\mathbf{x}], \mathbf{Q}_{[\mathbf{p}]}) = [\mathbf{f}]_{[\mathbf{x}]}^*$ . La Fig.3.4 illustre des fonctions robustes d'inclusion.

[I]RMOA II élimine toutes les boîtes que sont dominées par les pires cas trouvés. Les mieux pires cas  $\mathbf{y}^*$ , trouvés dès la première itération, sont enregistrés dans la file  $\mathbf{Q}_{\mathbf{Y}^*}$ . Maintenant, on considère une boîte  $[\mathbf{x}]$  n'importe lequel. Le pire cas de  $[\mathbf{x}]$  est obtenu de (3.28) et (3.29). Donc,

$$[\mathbf{x}] \text{ est éliminé} \Leftrightarrow \exists \mathbf{y}^* \in \mathbf{Q}_{\mathbf{Y}^*}, \mathbf{y}^* \preceq [\mathbf{f}]_{[\mathbf{x}]}^*. \quad (3.32)$$

À la fin du procédé de recherche de [I]RMOA II,  $\mathbf{Q}_{\mathbf{Y}^*}$  est composé par les pires cas non dominés de tous les itérations. Ils composent la frontière de Pareto robuste.

[I]RMOA II est décrit dans la Table 3.7. Les paramètres de l'algorithme sont déjà présentés, sauf la file  $\mathbf{Q}_{\mathbf{X}^*}$  qui enregistre les centres des boîtes solutions, et la notation  $\mathbf{Q}_{[\mathbf{p}]} \{1\}$ , où le nombre 1 signifie la première position de la file  $\mathbf{Q}_{[\mathbf{p}]}$ .

Table 3.7: Algorithme [I]RMOA II.

entrée	sortie
$([\mathbf{X}], [\mathbf{P}], \varepsilon_{[\mathbf{x}]}, \varepsilon_{[\mathbf{p}]})$	$(\mathbf{Q}_{[\mathbf{x}]}, \mathbf{Q}_{\mathbf{X}^*}, \mathbf{Q}_{\mathbf{Y}^*})$
1	Initialiser $\mathbf{Q}_{\mathbf{X}^*}$ , $\mathbf{Q}_{\mathbf{Y}^*}$ et $\mathbf{Q}_{[\mathbf{y}]}$ avec $\emptyset$ .
2	Traite contraintes(entrée: $[\mathbf{X}], [\mathbf{P}], \varepsilon_{[\mathbf{x}]}, \varepsilon_{[\mathbf{p}]}$ ; sortie: $\mathbf{Q}_{[\mathbf{x}]}, \mathbf{Q}_{[\mathbf{p}]}$ )-Table 3.6.
3	Si $\mathbf{Q}_{[\mathbf{x}]}$ est vide, alors aller au <i>Pas 13</i> , sinon défiler $[\mathbf{x}]$ de $\mathbf{Q}_{[\mathbf{x}]}$ .
4	Appeler MIF( $[\mathbf{x}], \mathbf{Q}_{[\mathbf{p}]}$ ), mettre le résultat en $[\mathbf{f}]_{[\mathbf{x}]}^*$ .
5	Si existe un $\mathbf{y}$ en $\mathbf{Q}_{\mathbf{Y}^*}$ tel que $\mathbf{y} \preceq [\mathbf{f}]_{[\mathbf{x}]}^*$ alors aller au <i>Pas 3</i> .
6	Appeler MIF( $\mathbf{c}([\mathbf{x}]), \mathbf{Q}_{[\mathbf{p}]}$ ), mettre le résultat en $[\mathbf{f}]_{\mathbf{x}}^*$ .
7	Si existe un $\mathbf{y}$ en $\mathbf{Q}_{\mathbf{Y}^*}$ tel que $\mathbf{y} \preceq [\mathbf{f}]_{\mathbf{x}}^*$ ,
8	si $w([\mathbf{x}]) > \varepsilon_{[\mathbf{x}]}$ , alors bissecter $[\mathbf{x}]$ , enfiler les résultats à $\mathbf{Q}_{[\mathbf{x}]}$ et aller au <i>Pas 3</i> ; sinon enfiler $[\mathbf{f}]_{[\mathbf{x}]}^*$ à $\mathbf{Q}_{[\mathbf{y}]}$ et $[\mathbf{x}]$ en $\mathbf{Q}_{[\mathbf{x}]}$ .
9	Supprimer tout le $\mathbf{y}$ en $\mathbf{Q}_{\mathbf{Y}^*}$ tel que $[\overline{\mathbf{f}}]_{\mathbf{x}}^* \preceq \mathbf{y}$ et les respectifs éléments en $\mathbf{Q}_{\mathbf{X}^*}$ .
10	Enfiler $[\overline{\mathbf{f}}]_{\mathbf{x}}^*$ à $\mathbf{Q}_{\mathbf{Y}^*}$ et $\mathbf{c}([\mathbf{x}])$ à $\mathbf{Q}_{\mathbf{X}^*}$ .
11	Supprimer tout le $[\mathbf{y}]$ en $\mathbf{Q}_{[\mathbf{y}]}$ tel que $[\mathbf{f}]_{\mathbf{x}}^* \preceq [\mathbf{y}]$ et les respectifs boîtes en $\mathbf{Q}_{[\mathbf{x}]}$ .
12	Aller au <i>Pas 3</i> .
13	Retourner $\mathbf{Q}_{[\mathbf{x}]}$ , $\mathbf{Q}_{\mathbf{X}^*}$ et $\mathbf{Q}_{\mathbf{Y}^*}$ .

Références: Dao [27].

### 3.6 - Algorithme Évolutionnaire Multi-Objectif Robuste

[I]RMOEA est un algorithme stochastique que cherche trouver directement l'ensemble optimal  $\mathbf{X}^*$ . La structure algorithmique du [I]RMOEA est similaire à d'un Algorithme Génétique en général, mais internement, les variables sont écrites en utilisant le format réel et intervalles. En conséquence, les processus qui manipulaient ces variables sont aussi adaptés pour traiter les deux sortes de données. Avant de décrire le [I]RMOEA, il est expliqué quelques notions importantes pour mieux lui comprendre, comment la formation de la population, la méthode de sélection, d'élitisme et de niche utilisés.

Considère la notation  $\mathbf{x}^i$  pour un individu  $i$ , qui dans l'optimisation est le paramètre de recherche. En génération  $t$ , la population  $\mathbf{pop}_t$  est :

$$\mathbf{pop}_t = \{\mathbf{x}^1, \dots, \mathbf{x}^i, \dots, \mathbf{x}^{n_{pop}}\}, \quad (3.34)$$

soient  $\mathbf{pop}_t(i)$  l'accès au  $i$ -ième individu. La performance de  $\mathbf{pop}_t$  est :

$$\mathbf{des}_t = \{\overline{[\mathbf{f}]_{\mathbf{x}^1}}^*, \dots, \overline{[\mathbf{f}]_{\mathbf{x}^i}}^*, \dots, \overline{[\mathbf{f}]_{\mathbf{x}^{n_{pop}}}}^*\}. \quad (3.35)$$

Par analogie,  $\mathbf{des}_t(i)$  représente la performance du  $i$ -ième individu. La Table 3.8 exhibe les détails pour trouver  $\mathbf{des}_t$ .

Le traitement des contraintes est fait de la même façon que celle présentée pour l'[I]RMOA II en (3.30). La différence c'est que [I]RMOA II a besoin aussi d'évaluer le vecteur  $\mathbf{x}$  dans les tests d'inclusion. Ainsi,

$$[t](\mathbf{x}) = \begin{cases} 1, & \text{si } [\mathbf{g}](\mathbf{x}, [\mathbf{p}]) \subseteq [-\infty, 0], \\ 0, & \text{si } [\mathbf{g}](\mathbf{x}, [\mathbf{p}]) \subset [0, \infty], \text{ et} \\ [0, 1], & \text{autrement.} \end{cases} \quad (3.36)$$

La procédure utilisée pour choisir les individus dans la population est la méthode de sélection du type *Roulette*. Dans les problèmes mono-objectif, cette méthode attribue à chaque individu une valeur de pourcentage, selon l'évaluation obtenue de la fonction objectif, qui représente la probabilité d'individu d'être choisi. Dans les méthodes multi-objectif, la méthode Roulette est peu efficace parce qu'il a, normalement, des objectifs qui sont contradictoires. Donc, dans cette thèse, il est développé une agrégation des objectifs que ne favorise pas directement un objectif spécifique. Ainsi, la probabilité reste dépendant de la valeur obtenu pour l'agrégation. La *performance utile*  $\mathbf{des}'_t(i)$  de chaque individu  $i$  est mesurée comme :

$$\mathbf{des}'_t(i) = \frac{1}{f_{nicho}^i} + \frac{1}{f_{front}^i}. \quad (3.37)$$

Soient  $f_{niche}^i$  la valeur de performance selon la quantité d'individus que participent de la même niche que l'individu  $i$ , et  $f_{front}^i$  selon la frontière de dominance où l'individu  $i$  est situé. Par exemple, si l'individu  $i$  appartient à la première frontière (de Pareto) alors  $f_{front}^i=1$  ; si il appartient à la deuxième frontière alors  $f_{front}^i=2$ , etc. La méthode Roleta utilise la valeur  $\mathbf{des}'_t(i)/\sum_{j=1}^{n_{pop}} \mathbf{des}'_t(j)$  de probabilité pour chaque individu  $i$ .

La codification de l'individu, la méthode de croisement et la méthode de mutation employés dans [I]RMOEA sont classifiées comment classiques. La codification est binaire, le croisement est réalisé avec un point de coupe et la mutation est faite avec le changement de bits, selon une probabilité  $p_m$ . La technique de élitisme consiste en ajouter les populations des générations  $t$  et  $t + 1$ , et choisir les premières  $n_{pop}$  individus des premières frontières.

Les méthodes de niche ont la finalité d'obtenir et maintenir un niveau de diversité dans les populations à travers de l'encouragement la formation de sous-populations de solutions stables. Ainsi, les niches permettent aux algorithmes génétiques de rechercher des régions en parallèle et donc d'éviter la convergence anticipée de l'algorithme à une région optimale local. Plusieurs méthodes de niche ont été reportées dans la littérature. Dans ce travail, est introduit la technique de niche NB[I] (Niches par Bisection d'Intervalles). NB[I] sépare les niches naturellement sans avoir besoin d'exécuter calculs de distance entre les solutions, qui est un inconvénient normalement cité dans les travaux scientifiques. NB[I] commence par encadrer toutes les solutions, noté pour la performance individuelle  $\overline{[f]}_{\mathbf{x}^i}^*$ , dans une boîte  $[\mathbf{F}]_0$  sur l'espace des objectifs. Ainsi, pour chaque individu, on a l'inclusion suivante :

$$\overline{[f]}_{\mathbf{x}^i}^* \subseteq [\mathbf{F}]_0, \quad [\mathbf{F}]_0 = [F_1, \overline{F}_1] \times \dots \times [F_j, \overline{F}_j] \times \dots \times [F_{n_f}, \overline{F}_{n_f}], \quad (3.38)$$

avec :

$$\underline{F}_j = \min_{i=1, \dots, n_{pop}} \overline{[f]}_{\mathbf{x}^i}^* \quad \text{et} \quad \overline{F}_j = \max_{i=1, \dots, n_{pop}} \overline{[f]}_{\mathbf{x}^i}^*. \quad (3.39)$$

La boîte  $[\mathbf{F}]_0$  est la niche initiale. Dans la prochaine iteration,  $[\mathbf{F}]_0$  est bisecté en produisant les deux boîtes :

$$[\mathbf{F}]_1 = [F_1] \times \dots \times [F_j, \frac{F_j + \overline{F}_j}{2}] \times \dots \times [F_{n_f}], \quad (3.40)$$

$$[\mathbf{F}]_2 = [F_1] \times \dots \times [\frac{F_j + \overline{F}_j}{2}, \overline{F}_j] \times \dots \times [F_{n_f}]. \quad (3.41)$$

Ainsi, la classification de  $\mathbf{x}^i$  est réglée par :

$$\begin{cases} \overline{\mathbf{f}}_{\mathbf{x}^i}^* \in [\mathbf{F}]_1, & \text{si } \overline{f_j}_{\mathbf{x}^i}^* \leq \frac{F_j + \overline{F}_j}{2}, \\ \overline{\mathbf{f}}_{\mathbf{x}^i}^* \in [\mathbf{F}]_2, & \text{autrement.} \end{cases} \quad (3.42)$$

Après la classification de toutes les solutions, s'il y a une niche vide, alors elle est éliminée. Le procédé de bisection continue jusqu'à un *nombre maximale de cycles de bisection*  $r_{bis}^{max}$ , soient un cycle de bisection l'exécution d'une seule bisection en chaque boîte qui était disponible dès le début de l'itération. Sur le texte principal du mémoire, la Fig.3.6 (page 131) et la Fig.3.7 (page 133) présentent quelques itérations réalisées sur un exemple générique. La Table 3.9 détaille la méthode NB[I]. Après les définitions, le [I]RMOEA est présenté dans la Table 3.11 et ses paramètres dans la Table 3.12.

Table 3.11: Algorithme [I]RMOEA.

<b>entrée</b> ( $n_{bits}, n_f, n_{pop}, n_{gen}, p_c, p_m, [\mathbf{X}], [\mathbf{P}], \varepsilon_{[p]}$ ) <b>sortie</b> ( $\mathbf{pop}_t, \mathbf{des}_t$ )	
1	Initialiser $t$ avec 0.
2	Traiter contraintes(entrée: $[\mathbf{X}], [\mathbf{P}], \varepsilon_{[x]}, \varepsilon_{[p]}$ ; sortie: $\mathbf{Q}_{[x]}, \mathbf{Q}_{[p]}$ )-Table 3.6.
3	Initialiser $\mathbf{pop}_t$ avec $n_{bits}, n_f, n_{pop}$ et $\mathbf{Q}_{[x]}$ .
5	Calculer $\mathbf{des}_t = \overline{\mathbf{f}}_{\mathbf{x}}^*$ , $\mathbf{x} \in \mathbf{pop}_{aux}$ - Equation (3.28) et Equation (3.29).
6	Classifier les frontières et calculer $f_{front}^i$ .
7	Si $t \geq n_{gen}$ aller au <i>Pas 15</i> .
8	Ajouter 1 à $t$ .
9	Calculer $f_{niche}^i$ - NBI.
10	Calculer $\mathbf{des}'_t$ - Equation (3.37).
10	Réaliser <i>sélection</i> - probabilité $\mathbf{des}'_t(i) / \sum_{j=1}^{n_{pop}} \mathbf{des}'_t(j)$ .
11	Réaliser <i>croisement</i> et <i>mutation</i> selon les probabilités $p_c$ et $p_m$ . Composer la population $\mathbf{pop}_{aux}$ .
12	Remplacer $\forall \mathbf{x} \in \mathbf{pop}_{aux}$ tel que $[\mathbf{g}](\mathbf{x}, \mathbf{Q}_{[p]}) \subset [0, \infty]$ par autre généré par événement aléatoire en $\mathbf{Q}_{[x]}$ .
13	Calculer $\mathbf{des}_{aux} = \overline{\mathbf{f}}_{\mathbf{x}}^*$ , $\mathbf{x} \in \mathbf{pop}_{aux}$ - Equation (3.28) et Equation (3.29).
14	Réaliser élitisme en considérant $\mathbf{des}_{t-1} \cup \mathbf{des}_{aux}$ , composer la population $\mathbf{pop}_t$ et aller au <i>Pas 7</i> .
15	Retourner $\mathbf{pop}_t$ et $\mathbf{des}_t$ .

Table 3.12: Paramètres du [I]RMOEA.

Symboles	Description	Type
$n_{bits}$	nombre de bits de la codification	$\mathbb{Z}$
$n_{gen}$	nombre de générations	$\mathbb{Z}$
$n_{pop}$	nombre d'individus	$\mathbb{Z}$
$n_f$	nombre de fonctions objectif	$\mathbb{Z}$
$f_{front}^i$	performance d'individu $i$ selon sa frontière	$\mathbb{Z}$
$f_{niche}^i$	performance d'individu $i$ selon sa niche	$\mathbb{Z}$
$\mathbf{pop}_t$	population en $n_{gen} = t$	Matrice: $\mathbb{R}^{n_f}$
$\mathbf{des}_t$	performance de $\mathbf{pop}_t$	Matrice: $\mathbb{R}^{n_f}$
$\mathbf{pop}_{aux}$	population auxiliaire	Matrice: $\mathbb{R}^{n_f}$
$\mathbf{des}_{aux}$	performance de $\mathbf{pop}_{aux}$	Matrice: $\mathbb{R}^{n_f}$

Références: Deb [32], et Deb et al. [34].

### 3.7 - Métriques de Mesure des Performances Révisées

Dans cette section est présenté une technique, appelée MB[I] (Métrique par Bisection d'Intervalles), pour mesurer l'homogénéité des images des solutions sur la frontière de Pareto. La structure algorithmique est similaire à celle décrite pour la technique NB[I]. La métrique MB[I] commence par encadrer toutes les solutions, selon ses valeurs de performance  $\mathbf{des}_t$  :

$$\mathbf{des}_t = \{ \mathbf{f}_{\mathbf{x}^1}, \dots, \mathbf{f}_{\mathbf{x}^i}, \dots, \mathbf{f}_{\mathbf{x}^{n_{pop}}} \}, \quad i = \{1, \dots, n_{pop}\} \quad (3.43)$$

soient  $\mathbf{f}_{\mathbf{x}^i} \in \mathbb{R}^{n_f}$  et l'accès à  $j$ -ième fonction objectif d'individu  $\mathbf{x}^i$  est fait par  $f_{j\mathbf{x}^i}$ . Ainsi, pour chaque individu, on a l'inclusion suivante :

$$\mathbf{f}_{\mathbf{x}^i} \subseteq [\mathbf{F}]'_0, \quad [\mathbf{F}]'_0 = [\underline{E}'_1, \overline{F}'_1] \times \dots \times [\underline{E}'_j, \overline{F}'_j] \times \dots \times [\underline{E}'_{n_f}, \overline{F}'_{n_f}], \quad (3.44)$$

avec :

$$\underline{E}'_j = \min_{i=1, \dots, n_{pop}} f_{j\mathbf{x}^i} \quad \text{e} \quad \overline{F}'_j = \max_{i=1, \dots, n_{pop}} f_{j\mathbf{x}^i}. \quad (3.45)$$

Un exemple de la construction de la boîte  $[\mathbf{F}]'_0$  est montré dans la Fig.3.8, itération 0. Le prochain pas est de bissecter  $[\mathbf{F}]'_0$ . La bisection est faite sur un individu  $k$  interne à  $[\mathbf{F}]'_0$  et une fonction objectif  $j$  choisi un événement

aléatoire,

$$[\mathbf{F}]'_1 = [F_1] \times \dots \times [E_j, f_{j_{\mathbf{x}^k}}] \times \dots \times [F_{n_f}], \quad (3.46)$$

$$[\mathbf{F}]'_2 = [F_1] \times \dots \times [f_{j_{\mathbf{x}^k}}, \overline{F}_j] \times \dots \times [F_{n_f}]. \quad (3.47)$$

Pour mieux comprendre la première bisection, on observe la Fig.3.8 (page 136), iteration 1. Les individus  $i$  dans la boîte  $[\mathbf{F}]'_0$  sont séparés en considérant l'objectif  $j$ , selon la règle :

$$\begin{cases} \mathbf{f}_{\mathbf{x}^i} \in [\mathbf{F}]_1, & \text{si } f_{j_{\mathbf{x}^i}} \leq f_{j_{\mathbf{x}^k}}, \\ \mathbf{f}_{\mathbf{x}^i} \in [\mathbf{F}]_2, & \text{autrement.} \end{cases} \quad (3.48)$$

À la suite, les bisections continuent jusqu'au pas où toutes les boîtes qui lient les points sont vides, conformément à ce qui est montré sur la Fig.3.8 (page 136), itération 11. À la fin du procédé de bisection, toutes les boîtes sont mémorisées dans la file  $\mathbf{Q}_{[\mathbf{F}]}$ , et on calcule le volume  $v([\mathbf{F}])$  de chacune, la métrique  $MB[\mathbf{I}]$  est définie comme l'écart type  $\sigma_{v([\mathbf{F}])}$ , ou formellement,

$$MB[\mathbf{I}] = \sigma_{v([\mathbf{F}])}. \quad (3.49)$$

Plus  $\sigma_{v([\mathbf{F}])}$  est petit, meilleure sera l'homogénéité de la frontière de Pareto. Alternativement, on peut utiliser une valeur normalisée  $MB[\mathbf{I}]_{norm}$  :

$$MB[\mathbf{I}]_{norm} = \frac{MB[\mathbf{I}] - MB[\mathbf{I}]_{front}}{MB[\mathbf{I}]_{max} - MB[\mathbf{I}]_{front}}, \quad MB[\mathbf{I}]_{max} \neq MB[\mathbf{I}]_{front}, \quad (3.50)$$

soient  $MB[\mathbf{I}]_{front}$  la valeur de  $MB[\mathbf{I}]$  calculée pour un ensemble solution de Pareto connu, et  $MB[\mathbf{I}]_{max}$  la valeur maximale trouvée pour la métrique après l'évaluation d'autres ensembles solution.

Cette section a réalisé aussi un comparatif entre les autres métriques présentés sur le chapitre dernier. Les résultats situent  $MB[\mathbf{I}]$  comme une méthode efficace pour évaluer la qualité de la distribution de points sur la frontière de Pareto.

---

## Chapitre 4 - Validation Numérique des Algorithmes pour l'Optimisation Robuste

Dans ce chapitre, sont présentés des tests de validation pour les algorithmes [I]RMOA I, [I]RMOA II et [I]RMOEA. La validation a été réalisée par moyen de fonctions tests. Les expressions mathématiques des fonctions tests ont été formulées selon la philosophie du scénario du pire cas. Il est important d'observer que les méthodes proposées dans ce travail sont lentes et un peu précises si comparées avec des méthodes spécifiques pour un problème particulier. Par exemple, si le problème robuste en question est continu et convexe, alors il y a des *méthodes de points intérieurs*, basés en l'information du gradient, pour résoudre ce genre de problème rapidement, précisément et en considérant plusieurs centaines de variables. Par contre, les algorithmes proposés ont été développés pour traiter des problèmes d'optimisation multi-objectif génériques, c'est à dire, qu'ils soient: a) de recherche globale ; b) garantis pour envelopper la frontière robuste ([I]RMOA I et [I]RMOA II) ; b) n'utilisent pas de dérivées ; c) ne demandent pas de probabilités pour le calcul des incertitudes ; et d) ne nécessitant pas de convexité de la Frontière de Pareto.

*Références: Kouvelis et Yu [75], et Boyd et Vandenberghe [13]*

### 4.1 - Méthodologie

En partant de l'idée de valider les méthodes d'optimisation proposées, ont été formulés cinq problèmes tests qui présentent quelques caractéristiques différentes comme la non convexité et non continuité de la frontière de Pareto robuste. Les problèmes choisis sont des versions robustes adaptées de problèmes tests multi-objectif traditionnelles, et ils prennent en compte deux objectifs pour faciliter l'interprétation graphique des résultats. Donc, la validation consiste d'analyse de tests expérimentaux.

### 4.2 - Fonctions Test

La définition de problèmes robustes est dépendant d'interprétation d'action des incertitudes sur le système d'optimisation, comme cela a été discuté antérieurement. Dans ce travail, l'incertitude a été considérée un paramètre borné et différent à la variable de projet. Les fonctions tests robustes utilisées sont présentées à la suite. Dans le texte principal du rapport se trouvent aussi les expressions originales de chaque problème et un graphique confrontant les images résultantes des deux cas.

La fonction SCH2 (Fonction Schaffer 2 - Equation 4.2, Fig.4.1) se caractérise par utiliser q'une seule variable, par avoir la frontière de Pareto non continue et par avoir simples occurrences de  $x$  et  $\mathbf{p}$ . Ce dernier aspect est important pour la qualité des résultats des fonctions d'inclusion. Quand il y a plusieurs occurrences, la fonction d'inclusion produit une image plus pessimiste, qui est indésirable. Dans ce cas, l'incertitude peut être classifié comme quelque type d'imprécision de  $x$ . L'ensemble solution robuste est  $\mathbf{X}^* \in \{[1, 2] \cup [4, 5]\}$ . Le pire cas se trouve en  $p_1 = 0.1$  par  $f_1$  et en  $\mathbf{p} = \{-0.1, 0.1\}$  por  $f_2$ . Comme conséquence  $\mathbf{Y}^* \not\subset \mathbf{f}(x, \mathbf{p})$ .

$$\min_{x \in [-5, 10]} \max_{\mathbf{p} \in [-0.1, 0.1]} f_1(x, \mathbf{p}) = \begin{cases} -(x + p_1), & \text{se } x \leq 1 \\ (x + p_1) - 2, & \text{se } 1 < x \leq 3 \\ 4 - (x + p_1), & \text{se } 3 < x \leq 4 \\ (x + p_1) - 4, & \text{se } x > 4 ; \end{cases} \quad (4.2)$$

$$f_2(x, \mathbf{p}) = ((x + p_1) - 5)^2 + p_2.$$

La fonction FON (Fonseca et Fleming - Equation 4.4, Fig.4.2) se caractérise par ses  $n$  variables et deux objectifs, par avoir la frontière de Pareto continue, non convexe. Aussi dans ce cas, l'incertitude peut être interprété comment une sorte d'imprécision de paramètre. Le pire cas se trouve en  $p = 1.3$ , quand  $x_i = 1/\sqrt{n}$  et  $x_i = -1/\sqrt{n}$ , par  $f_1$  et  $f_2$ , respectivement.

$$\min_{\mathbf{x} \in [-4, 4]} \max_{p \in [1.1, 1.3]} f_1(\mathbf{x}, p) = 1 - e^{-\sum_{i=1}^n (x_i - \frac{p}{\sqrt{n}})^2} ; \quad (4.4)$$

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

La fonction ZDT1 (Fonction Zitzler, Deb et Thiele 1 - Equation 4.7, Fig.4.3) se caractérise par ses  $n$  variables, par avoir la

$$\min_{\mathbf{x} \in [0, 1]} \max_{\mathbf{p} \in [0, 0.05]} f_1(\mathbf{x}, \mathbf{p}) = x_1 + p_1 ;$$

$$f_2(\mathbf{x}, \mathbf{p}) = h'(\mathbf{x}) \times h(f_1, h', \mathbf{p}). \quad (4.7)$$

$$\text{soient :} \quad h'(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i,$$

$$h(f_1, h', \mathbf{p}) = 1 - \sqrt{f_1/h'} + p_2.$$

sa frontière de Pareto continue et convexe, et par ses multiples occurrences de  $x$  et  $\mathbf{p}$ . Dans ce cas, le paramètre d'incertitude représente un type d'imprécision de  $x_1$  en  $f_1$ , et l'idée que  $f_2$  est'une approximation de sa

valeur correcte. Le domaine de la région robuste est  $x_1 \in [0, 1]$  et  $x_i = 0$ ,  $i = 2, \dots, n$ .

La fonction ZDT2 (Fonction Zitzler, Deb et Thiele 2 - Equation 4.9, Fig.4.4) se caractérise par avoir contraintes, par ses  $n$  variables, par avoir la frontière de Pareto non continue et non convexe, et par avoir multiples occurrences de  $x$  et  $\mathbf{p}$ . Aussi dans ce cas, la fonction  $f_2$  est une approximation de sa valeur correcte. Le domaine de la région robuste est  $x_1 \in [0, 1]$  et  $x_i = 0$ ,  $i = 2, \dots, n$ .

$$\begin{aligned} \min_{\mathbf{x} \in [0,1]} \max_{\mathbf{p} \in [-0.05, 0.05]} \quad & f_1(\mathbf{x}, p) = x_1 ; \\ & f_2(\mathbf{x}, p) = h'(\mathbf{x}, p) \times h(f_1, h', p). \\ \text{s.c.} \quad & g(\mathbf{x}) = \sqrt{(x_1 - 0.7)^2 + x_2^2 + \dots + x_n^2} > 0.1 \quad (4.9) \end{aligned}$$

$$\begin{aligned} \text{soient :} \quad & h'(\mathbf{x}, p) = 1 + \frac{9}{n-1} (\sum_{i=2}^n x_i) + p, \\ & h(f_1, h', p) = 1 - (f_1/h')^2 + p. \end{aligned}$$

La fonction ZDT3 (Fonction Zitzler, Deb et Thiele 3 - Equation 4.11, Fig.4.5) se caractérise par ses  $n$  variables, par avoir la frontière de Pareto non continue, par avoir l'espace des objectifs multimodaux, et par ses multiples occurrences de  $x$  et  $\mathbf{p}$ . Dans ce cas, le paramètre d'incertitude représente l'idée que la fonction  $f_2$  est une approximation de sa valeur correcte. Le domaine de la région robuste est  $x_1 \in [0, 1]$  et  $x_i = 0$ ,  $i = 2, \dots, n$ .

$$\begin{aligned} \min_{\mathbf{x} \in [0,1]} \max_{\mathbf{p} \in [-0.1, 0.1]} \quad & f_1(\mathbf{x}, ) = x_1 ; \\ & f_2(\mathbf{x}) = h'(\mathbf{x}, p) \times h(f_1, h', p). \end{aligned} \quad (4.11)$$

$$\begin{aligned} \text{soient :} \quad & h'(\mathbf{x}, p) = 1 + \frac{9}{n-1} ((\sum_{i=2}^n x_i) + p), \\ & h(f_1, h', p) = 1 - \sqrt{f_1/h'} - (f_1/h') \sin(10\pi f_1). \end{aligned}$$

*Références: Barichard et Hao [6], Coello et Sierra [21], Deb et al. [34], Deb [32], Schaffer [103], Fonseca et Fleming [43], Deb [31], et Zitzler et al. [132]*

### 4.3 - Tests et Résultats

Dans les tests, il a été observé comment les algorithmes se comportent quand ses paramètres changent de valeurs. Les critères d'évaluation ont été basés en approximations de l'ensemble solution robuste  $\mathbf{X}^*$  et son image  $\mathbf{Y}^*$ . Donc, pour aider ces comparaisons, ont été définis un ensemble solution discrète  $\mathbf{X}_d^* \subset \mathbf{X}^*$  et sa image  $\mathbf{Y}_d^* \subset \mathbf{Y}^*$  comme les ensembles de *référence* à obtenir pour chaque fonction test ( $\mathbf{X}_d^*$  a été trouvé en testant toutes les solutions d'un domaine discret), et les ensembles  $\mathbf{X}_s^*$  et  $\mathbf{Y}_s^*$  représentent les résultats obtenus des algorithmes proposés.

Les algorithmes sont différents, donc les paramètres à tester sont aussi un peu variés. Ainsi, les résultats sont présentés en considérant les exécutions réalisées pour chaque algorithme de façon séparée. Avant les résultats, un sommaire sur la Table 4.1 présent quelques valeurs fixées des problèmes tests.

Table 4.1: Sommaire du problème tests pour l'optimisation robuste.

Fonction	$\mathbf{f}(\mathbf{x}, \mathbf{p})$ , $\mathbf{g}(\mathbf{x}, \mathbf{p})$	$[\mathbf{X}] \in \mathbb{IR}^{n_x}$	$[\mathbf{P}] \in \mathbb{IR}^{n_p}$	$n_x$	$n_p$
SCH2	(4.2)	$[-5, 10]$	$[-\mathbf{0.1}, \mathbf{0.1}]$	1	2
FON	(4.4)	$[-\mathbf{4}, \mathbf{4}]$	$[1.1, 1.3]$	2	1
ZDT1	(4.5) et (4.7)	$[\mathbf{0}, \mathbf{1}]$	$[\mathbf{0}, \mathbf{0.05}]$	2	2
ZDT2	(4.5) et (4.9)	$[\mathbf{0}, \mathbf{1}]$	$[-0.05, 0.05]$	2	1
ZDT3	(4.5) et (4.11)	$[\mathbf{0}, \mathbf{1}]$	$[-0.1, 0.1]$	2	1

#### 4.3.1 - [I]RMOA I

La performance et l'efficacité du [I]RMOA I dépendent de ses paramètres de précision  $\varepsilon_{\mathbf{x}\#}$  et  $\varepsilon_{[\mathbf{x}]}$ . Donc, les tests consistent en observer comment l'algorithme se comporte quand  $\varepsilon_{\mathbf{x}\#}$  et  $\varepsilon_{[\mathbf{x}]}$  changent leurs valeurs. Pour cet algorithme trois résultats de tests sont présentés : a) la quantité d'éléments de  $\mathbf{K}^-$  et  $\mathbf{K}^+$ , qui définissent l'enveloppe sur la frontière robuste ; b) l'effort calculatoire pour évaluer les échantillons obtenus de la fonction *grid* et pour compléter toute la simulation (ces efforts ont été mesurés en nombre de calculs de fonctions objectif) ; et c) la qualité des enveloppes sur  $\mathbf{Y}^*$ .

Pour réaliser les tests, ont été utilisés plusieurs valeurs pour  $\varepsilon_{\mathbf{x}\#}$  et  $\varepsilon_{[\mathbf{x}]}$ . En considérant que la notation  $\varepsilon_{\mathbf{x}\#} = [0.02 : 0.02 : 0.1]$  pour représenter la structure de données  $\varepsilon_{\mathbf{x}\#} = \{0.02, 0.04, \dots, 0.1\}$ , la table à la suite décrite de valeur de paramètres de précision testés.

Fonction	$\varepsilon_{\mathbf{x}\#}$	$\varepsilon_{[\mathbf{x}]}$
SCH2	[0.02 : 0.02 : 0.1]	[0.02 : 0.02 : 0.1]
FON	[0.05, 0.1, 0.2 : 0.2 : 1]	[0.2 : 0.2 : 1]
ZDT1	[0.02 : 0.01 : 0.1]	[0.02 : 0.01 : 0.1]
ZDT2	[0.02 : 0.02 : 0.1]	[0.01, 0.02 : 0.02 : 0.1]
ZDT3	[0.02 : 0.02 : 0.1]	[0.02 : 0.01 : 0.1]

L’enveloppe sur la frontière robuste est composée pour les éléments des ensembles discrets  $\mathbf{K}^-$  et  $\mathbf{K}^+$ . Ainsi, la quantité de ces éléments reflètent une augmentation de la précision des forme d’enveloppe. Les figures Fig.4.6, Fig.4.9, Fig.4.12, Fig.4.15 et Fig.4.18 confrontent  $\mathbf{K}^-$  et  $\mathbf{K}^+$ , en considérant 3 valeurs de  $\varepsilon_{[\mathbf{x}]}$ . En analysant les figures, il est constaté que l’échantillonnage, défini par  $\varepsilon_{\mathbf{x}\#}$ , est le facteur fondamental pour contrôler le taille les ensembles  $\mathbf{K}^-$  et  $\mathbf{K}^+$ .

[I]RMOA I se présente comme un algorithme “coûteux” en considérant la quantité de calcul de fonctions objectif dépensées pour résoudre tous les cas discutés. La question du coût calculatoire est montré dans les figures Fig.4.7, Fig.4.10, Fig.4.13, Fig.4.16 et Fig.4.19. L’effort principal se concentre sur les deux fonctions FPS (Table 3.3) et CSC (Table 3.4) quand elles essayent, en vain, trouver un minimiseur pour une candidat à  $\mathbf{K}^-$ .

Quelques résultats des enveloppes obtenues ont été exhibés dans les figures Fig.4.8, Fig.4.11, Fig.4.14, Fig.4.17 et Fig.4.20. Parfois, les figures ont été présentées des simulations qui ne réussissaient pas envelopper toute la frontière robuste comment en Fig.4.11(b) Fig.4.20(a). Les raisons pour lesquelles ça est arrivé sont dues aux fonctions d’inclusions utilisées, que ne sont pas fines, car de la multiplicité d’occurrences des paramètres dans la formulation, par exemple. Pour équilibrer l’effet de la multiplicité, le paramètre  $\varepsilon_{[\mathbf{x}]}$  doit être suffisant plus petit que  $\varepsilon_{\mathbf{x}\#}$ , au moins  $\varepsilon_{[\mathbf{x}]} < \varepsilon_{\mathbf{x}\#}/2$ , pour assurer que la précision de recherche soit mieux que la précision de l’échantillonnage.

En considérant les résultats, le [I]RMOA I a réussi envelopper les frontières robustes dès que les facteurs de précision sont réglés et que le vecteur de fonctions objectif d’inclusion soit fini. Le processus de recherche dans le [I]RMOA I est simple et utilise seulement deux paramètres. De l’autre côté, des inconvénients sont liés au grand effort dépensé pour finir la recherche par solutions, et au fait qu’il trouve une enveloppe sur  $\mathbf{Y}^*$  et non sur le  $\mathbf{X}^*$ . En fait, il y a des méthodes qui font l’inversion d’ensembles et trouvent  $\mathbf{X}^*$ . Mais, l’enveloppe sur  $\mathbf{Y}^*$  est aussi une donnée intéressant, car le [I]RMOA I peut devenir une méthode de validation pour autres méthodes. Ce sujet est traité dans la Section 4.4.

## 4.3.2 - [I]RMOA II

La performance et l'efficacité du [I]RMOA II dépendent de ses paramètres de précision  $\varepsilon_{[x]}$  et  $\varepsilon_{[p]}$ . Pour chaque fonction test,  $\varepsilon_{[p]}$  a été fixé, et ainsi, les tests consistent en observer comment l'algorithme se comporte quand  $\varepsilon_{[x]}$  change de valeur. Pour cet algorithme trois résultats de tests ont été présentés: a) la quantité de solutions trouvées ; b) l'effort calculatoire ; c) la comparaison entre  $\mathbf{X}_s^*$  et  $\mathbf{X}_d^*$  ; et d) la comparaison entre  $\mathbf{Y}_s^*$  et  $\mathbf{Y}_d^*$ . Les valeurs utilisées pour  $\varepsilon_{[x]}$  et  $\varepsilon_{[p]}$  sont montrés dans la table à la suite.

Fonction	$\varepsilon_{[x]}$	$\varepsilon_{[p]}$
SCH2	[0.1 : 0.1 : 1]	0.2
FON	[0.1 : 0.1 : 1]	0.025
ZDT1	[0.01 : 0.01 : 0.1]	0.02
ZDT2	[0.01 : 0.01 : 0.1]	0.02
ZDT3	[0.01 : 0.01 : 0.1]	0.02

Selon la Table 3.7, les produits de [I]RMOA II sont une file  $\mathbf{Q}_{[x]}$  qui contient toutes les boîtes qu'enveloppent les solutions robustes, et deux autres files  $\mathbf{Q}_{\mathbf{X}^*}$  et  $\mathbf{Q}_{\mathbf{Y}^*}$ , composées pour les points centraux de chaque élément de  $\mathbf{Q}_{[x]}$  et de ses images robustes, respectivement. Les figures Fig.4.21, Fig.4.24, Fig.4.27, Fig.4.30 et Fig.4.33 présentent comment le nombre de solutions augmente avec la précision de  $\varepsilon_{[x]}$ . En fait, à chaque itération ils sont réalisés des bisections sur toutes les boîtes enregistrées en  $\mathbf{Q}_{[x]}$ , et après les bisections, l'algorithme [I]RMOA II met à jour  $\mathbf{Q}_{[x]}$ ,  $\mathbf{Q}_{\mathbf{X}^*}$  et  $\mathbf{Q}_{\mathbf{Y}^*}$ . Par conséquent, à chaque itération la précision augmente et aussi la quantité de solutions. Aussi, les figures citées dernièrement ont montré l'effort calculatoire par valeur de précision. [I]RMOA II se présente aussi comment un algorithme coûteux, car dans toutes les itérations des évaluations sont doublées: sur les boîtes de  $\mathbf{Q}_{[x]}$  et sur ses centres  $\mathbf{Q}_{\mathbf{X}^*}$ .

Les figures Fig.4.22, Fig.4.25, Fig.4.28, Fig.4.31 et Fig.4.34 exhibent le confrontation entre les solutions discrètes  $\mathbf{X}_d^*$  et les solutions trouvés  $\mathbf{X}_s^*$ , en considérant deux valeurs de  $\varepsilon_{[x]}$ . Dans les cas, on observe que tous les éléments de  $\mathbf{X}_s^*$  ont atteint la précision déterminée par  $\varepsilon_{[x]}$ , et avec une évidente uniformité de la distribution. Cependant, [I]RMOA II est dit garanti. Pour obtenir plus solutions un nouveau cycle de bisections est requis.

Les figures Fig.4.23, Fig.4.26, Fig.4.29, Fig.4.32 et Fig.4.35 comparent les frontières robustes  $\mathbf{Y}_d^*$  et  $\mathbf{Y}_s^*$ , en considérant deux valeurs de  $\varepsilon_{[x]}$ . Dans les cas, on observe que la frontière trouvée  $\mathbf{Y}_s^*$  est uniforme et s'approche de  $\mathbf{Y}_d^*$  quand la valeur de  $\varepsilon_{[x]}$  se devenue plus petit.

En considérant les résultats, il peut être vérifier que le [I]RMOA II est une méthode multi-objectif globale, car il converge vers les régions optimales.

La proximité de  $\mathbf{X}_s^*$  à l'ensemble solution du problème est dépendant du paramètre de précision  $\varepsilon_{[x]}$ . Les tests ont montré que [I]RMOA II un peu sensible caractéristiques, qui normalement sont considérées source de difficultés, comme fonctions objectif non continues, non convexes et contraintes. Il y a aussi des inconvénients. Le nombre de variables et d'incertitudes peut laisser le procédé de bisection irréalisable, au moins qu'une stratégie optimiste soit utilisée (voir Fig.2.8). Le [I]RMOA II, comme toutes méthodes intervalles, est dépendent de la qualité de la fonction d'inclusion. Donc, il est tant et plus préférable que les fonctions objectif d'inclusion soient fines et minimales, et fondamental qu'elles soient convergentes et monotones.

### 4.3.3 - [I]RMOEA

La performance et l'efficacité du [I]RMOEA dépendent de plusieurs paramètres comme les autres algorithmes évolutionnaires en général. Pour simplifier les expérimentations, quelques paramètres ont été fixés, ils sont : a) probabilité de croisement  $p_c = 1$  ; b) probabilité de mutation  $p_m = 0.001$  ; et c) nombre maximal de cycles de bisections  $r_{bis}^{max} = 8$ . Les paramètres  $\varepsilon_{[p]}$  et  $\varepsilon_{TE}$  ont été définis pour chaque problème. Les tests ont évalué comment le [I]RMOEA répondre quand la taille de la population  $n_{pop}$  et le nombre de génération  $n_{gen}$  changent de valeur. Pour cet algorithme cinq résultats de tests ont été présentés: a) la convergence de l'algorithme, mesuré en nombre de frontières ; b) la comparaison entre  $\mathbf{X}_s^*$  et  $\mathbf{X}_d^*$ , via métrique TE ; c) l'homogénéité de l'ensemble Pareto, via métrique MB[I] ; d) la comparaison entre  $\mathbf{Y}_s^*$  et  $\mathbf{Y}_d^*$ , via métrique TE ; et e) l'effort calculatoire, mesuré en nombre de calculs de fonctions. Les valeurs utilisées pour  $n_{pop}$ ,  $n_{gen}$ ,  $\varepsilon_{[p]}$  et  $\varepsilon_{TE}$  sont listés dans la table présentée suivant.

Fonction	$n_{pop}$	$n_{gen}$	$\varepsilon_{[p]}$	$\varepsilon_{TE}$
SCH2	[10 : 10 : 100]	[10 : 10 : 50]	0.2	0.10
FON	[10 : 10 : 100]	[10 : 10 : 50]	0.02	0.05
ZDT1	[10 : 10 : 100]	[10 : 10 : 100]	0.01	0.01
ZDT2	[10 : 10 : 100]	[10 : 10 : 100]	0.01	0.01
ZDT3	[10 : 10 : 100]	[10 : 10 : 100]	0.01	0.01

Dans les algorithmes évolutionnaires, l'effort est normalement constant par itération principalement parce que la taille de la population de solutions ne change pas. C'est le cas du [I]RMOEA. La Fig.4.36 résume les valeurs d'effort par génération en considérant  $n_{pop}$  sur le abscisse, pour tous les problèmes tests. Le coût calculatoire du [I]RMOEA est le plus petit d'entre aux autres algorithmes proposés.

L'évolution de la quantité de frontières en fonction des générations est un indicateur de la convergence de l'IRMOEA vers l'ensemble  $\mathbf{Y}_s^*$ . Les figures Fig.4.37, Fig.4.41, Fig.4.45, Fig.4.49 et Fig.4.53 ont montré cette convergence. En fait, la convergence à la première frontière est importante, mais il est aussi important d'avoir diversité et uniformité dans l'ensemble optimal.

L'application de la métrique TE dans cette phase de tests a une double finalité: a) mesurer combien des solutions dans l'ensemble  $\mathbf{X}_s^*$  appartient  $\mathbf{X}_d^*$ ; et b) mesurer combien de solutions sont dupliquées. Le but en (b) a été de qualifier la diversité dans les ensembles produits. En utilisant la métrique TE, les figures Fig.4.38, Fig.4.42, Fig.4.46, Fig.4.50 et Fig.4.54 présentent que les meilleurs résultats ont été obtenus en utilisant des populations plus grandes et un nombre de générations plus grand. Mais, le nombre de duplicatas augmente aussi avec la taille de la population.

La métrique MB[I] a été développée pour mesurer l'uniformité des ensembles Pareto. La MB[I] est aussi un indicateur indirect de la qualité de la technique de niche NB[I], qui est le mécanisme responsable pour maintenir la diversité avant le procédé de sélection. Les figures Fig.4.39, Fig.4.43, Fig.4.47, Fig.4.51 et Fig.4.55 ont montré que les populations plus grandes ont eu plus de succès pour obtenir des ensembles plus uniformes.

Les résultats dans l'espace des objectifs ont été montrés sur les figures Fig.4.40, Fig.4.44, Fig.4.48, Fig.4.52 et Fig.4.56. L'uniformité de distribution des ensembles  $\mathbf{Y}_s^*$  sur la frontière robuste a été clairement meilleure quand ils ont été utilisés des populations de tailles plus grandes. Mais, il faut justifier le pourquoi. Dans les expérimentations, on a fixé le nombre maximal de cycles de bisection ( $r_{bis}^{max} = 8$ ) pour exécuter la technique NB[I]. Si  $r_{bis}^{max}$  définit le volume des niches et s'il y a plusieurs niches, alors les populations de tailles plus petites n'ont pas beaucoup de représentants pour appartenir à toutes les niches. Les niches vides sont éliminées dans les procédés du NB[I] amenant l'apparition de "trous" sur la frontière robuste.

Selon les résultats présentés pour l'IRMOEA, on considère que la méthode a obtenu succès pour trouver les frontières robustes en chaque cas, même quand l'IRMOEA a confronté les frontières robustes non continues et non convexes. Le traitement de contraintes par intervalles a été considéré efficace, car la classification a priori élimine toutes les régions irréalisables. Le coût d'exécution de l'algorithme a été bas quand on se compare avec les deux autres méthodes proposées. La convergence à la première frontière est arrivée avant la vingtième génération dans tous les cas étudiés. Ainsi, le critère pour arrêter l'IRMOEA a pu être considéré et finir l'exécution avant le nombre de générations fixé a priori.

*Références: Dao [27], et Soares [109]*

## 4.4 - Tests Complémentaires

Cette section a présenté des expérimentations spécifiques pour chaque méthode proposée. Pour le [I]RMOA I, il a été démontré que l'algorithme peut être utilisé pour valider d'autres méthodes développées pour l'optimisation multi-objectif robuste. Pour les [I]RMOA II et [I]RMOEA, les expérimentations réalisées ont validé la dépendance des algorithmes en relation à dimension du problème, c'est à dire lesquels ont des limitations pour résoudre les cas avec plusieurs variables.

### 4.4.1 - Le [I]RMOA I comment Outil de Validation

[I]RMOA I a été développé pour être une méthode de validation d'autres techniques construits pour traiter des problèmes robustes, formulés utilisant la philosophie du scénario du pire cas. Pour être apte à validation, le première pas est trouver, pour un problème connu, les paramètres de précision  $\varepsilon_{[x]}$  et  $\varepsilon_{x\#}$  que produisent une enveloppe correcte sur la frontière robuste. Dans cette phase, c'est le temps aussi de tester quelques fonctions objectif d'inclusion et choisir laquelle qui a le moins pessimisme. Supposons qu'une frontière robuste  $\mathbf{Y}_1^*$  et que un ensemble solution  $\mathbf{X}_1^*$  sont connus pour une valeur particulière d'incertitude  $\mathbf{p}_1$ . On considère aussi qu'ont été construits des fonctions objectif d'inclusion de "bonne" qualité, et qu'ont été trouvés une pair  $\varepsilon_{[x]}$  et  $\varepsilon_{x\#}$  qu'a résulté une analyse coût-bénéfice jugée satisfaisante. Encore, suppose que le scénario d'incertitude a changé complètement, c'est à dire, la valeur  $\mathbf{p}_1$  a changé pour  $\mathbf{p}_2$ , et probablement, la frontière robuste  $\mathbf{Y}_1^*$  ne sera pas consistant avec ce nouveau scénario. Ainsi, on denote  $\mathbf{Y}_2^*$  et  $\mathbf{X}_2^*$  les ensembles liés à  $\mathbf{p}_2$ . Dans l'espace de recherche, trois cas sont possibles : a)  $\mathbf{X}_1^* = \mathbf{X}_2^*$  ; b)  $\mathbf{X}_1^* \cap \mathbf{X}_2^* \neq \emptyset$ ,  $\mathbf{X}_1^* \neq \mathbf{X}_2^*$  ; c)  $\mathbf{X}_1^* \neq \mathbf{X}_2^*$ . Si l'action des incertitudes n'est pas claire dans les expressions des fonctions objectif, alors il peut être difficile, à priori, savoir lequel sera le cas de  $\mathbf{X}_2^*$ . Ainsi, en prenant en compte le scénario à  $\mathbf{p}_2$ , comment garantir qu'un ensemble  $\mathbf{Y}_2^*$  trouvé par une méthode quelconque est correct? Une possible réponse à cette question est l'application du [I]RMOA I pour envelopper  $\mathbf{Y}_2^*$ . Parce que, une fois qu'il a été vérifié pour résoudre le problème avec  $\mathbf{p}_1$ , il est garanti créer une autre enveloppe en utilisant  $\mathbf{p}_2$  (voir Théorème 2). À savoir, le [I]RMOA I utilisera les mêmes fonctions d'inclusion,  $\varepsilon_{[x]}$  et  $\varepsilon_{x\#}$  ; et un nouveau  $\mathbf{p}$ .

Les expérimentations ont consisté en confronter le [I]RMOA I et un algorithme stochastique adapté pour résoudre problèmes multi-objectif robustes.

Le RMOGA (Algorithme Génétique Multi-Objectif Robuste) utilise la configuration suivante : a) la codification binaire ; b) le croisement avec un point de coupe ; c) la mutation par changement de bits ; d) la sélection par Roulette (performance individuelle calculée par Equation 3.37) ; e) technique niche NB[I] (voir Sous-section 3.6.2) ; et f) l'élitisme a été défini comme le choix des premiers individus non dominés d'entre les deux dernières populations. Les fonctions tests choisies ont été ZDT3 et FON. La Fig.4.57 a présenté les enveloppes sur les frontières robustes correctes pour les cas ZDT3 et FON. Pour le RMOGA, le problème multi-objectif robuste a été défini comment une approximation stochastique de la philosophie du pire, soient :

$$\min_{\mathbf{x} \in \mathbf{X}} \max_{\mathbf{p}' \in \mathbf{P}'} \mathbf{f}(\mathbf{x}, \mathbf{p}'), \quad (4.12)$$

où  $\mathbf{P}'$  est un ensemble discrète d'échantillons sélectionné par événement aléatoire de  $\mathbf{P}$ . En observant (4.12), on note que la qualité de la frontière Pareto gérée dépend de la taille de  $\mathbf{P}'$ . Alors, le [I]RMOA I a été utilisé pour aider le RMOGA en trouver une taille de population que représente la frontière robuste. Dans les expérimentations, pour raisons de simplicité, le paramètre  $\mathbf{p}$  a été fixé, donc les enveloppes présentés sur la Fig.4.57 sont encore valides. Les tailles des populations testés ont été  $n_{pop} = \{5, 10, \dots, 50\}$ , et les autres paramètres ont été  $p_c = 1$ ,  $p_m = 0.0005$ ,  $n_{pop} = 50$  et  $n_{gen} = 40$ . Pour chaque cas, le RMOGA a été exécuté 10 fois, soient les résultats jugées plus intéressantes exhibés en Fig.4.58 et Fig.4.59. On observe que dans les deux cas, la taille inadéquat a géré des frontières robustes faux, soient le  $n_{pop} = 50$  la taille minimale to RMOGA produit des frontières de qualité acceptable.

#### 4.4.2 - [I]RMOA II × Dimension d'Espace de Recherche

Le but de cette sous-section est de présenter la limitation du [I]RMOA II en relation aux dimensions des paramètres d'un problème. Pour cet expérimentations, seulement la fonction ZDT3 a été utilisée. La précision pour la computation des incertitudes a été  $\varepsilon_{[\mathbf{p}]} = 0.01$  et l'espace de recherche analysée a été composé pour 3 et 4 variables. La précision  $\varepsilon_{[\mathbf{x}]} = 0.04$  a été employé pour limiter les bisections dans le processus de recherche. [I]RMOA II est une méthode garantie, alors les résultats obtenus ont été similaires à ceux présenté à la Sous-section 4.3.2, et donc ils ont été omis. Les simulations avec plusieurs variables pour les mêmes  $\varepsilon_{[\mathbf{x}]}$  et  $\varepsilon_{[\mathbf{p}]}$  cités plus haut ont été déclaré comment "irréalisables" pour l'équipement utilisé pour réaliser les tests (CPU Core Duo 1.73GHz/1.73GHz ; RAM 1024).

#### 4.4.3 - [I]RMOEA × Dimension d'Espace de Recherche

Dans cette sous-section, la batterie des tests vise à vérifier le comportement “moyen” du [I]RMOEA en fonction des dimensions des paramètres d’un problème. Pour réaliser les tests, seulement la version robuste de la fonction ZDT3 a été utilisée. Les espaces de recherche adoptés pour les simulations ont été 5, 10, ..., 30 variables, et la taille de la population a changé irrégulièrement de 15 à 300 individus, dépendant du cas. Le paramètre de précision d’incertitude a été fixé comme  $\varepsilon_{[p]} = 0.01$ . Ils ont été utilisés deux critères pour arrêter le [I]RMOEA: a) pour le nombre maximal de 200 générations ; et b) pour la convergence de la population dans la première frontière pendant 20 générations consécutives. En fait, les populations plus petites,  $n_{pop} \leq 100$ , ont la tendance de converger selon le critère d’arrêt (b). Trois tests ont été exécutés chaque fois. Les résultats jugés les plus intéressants sont présentés à la suite.

Les Fig.4.60 et Fig.4.61 ont exemplifié les résultats des frontières robustes pour les cas de 10 et 30 variables respectivement. On observe que la proximité de  $\mathbf{Y}_d^*$  dépend de la taille de la population. En analysant le cas plus critique (n=30 variables), les résultats ont été considérés satisfaisant, car le coût par génération est resté constant et la frontière a été obtenue. La Table 4.3 résume les résultats de cette section.

Table 4.3: [I]RMOEA - Résumé des tests complémentaires

n	$n_{pop}$	TE (%)	Dupliquées (%)	MB[I]
5	100	1.00	67.0	0.0040
10	100	3.00	52.0	0.0079
15	100	22.0	60.0	0.0244
20	200	1.50	62.5	0.0044
25	250	2.00	56.0	0.0040
30	300	3.33	47.7	0.0050

*Références: Soares et al [112], Deb et al. [34], et Soares et al. [110]*

---

## Chapitre 5 - Optimisation Robuste Multi-Objectif de Synthèse des Régulateurs PID

Normalement, le développement de fonctions d'inclusion part de l'expression analytique que décrit le problème. Dans le Chapitre 4, les fonctions test d'inclusion, classifiés comme fonctions d'inclusion naturelle, qui ont été obtenue par le remplacement direct des paramètres réels par son correspondant intervalle. Cependant, la construction de fonctions d'inclusion est une tâche plus complexe quand les situations problèmes réelles sont examinées. Il y a aussi des particularités pour traiter problèmes par intervalles, comme contraintes de singularités dans les opérations de division par exemple. Donc, ce chapitre a été dédié à traiter un problème d'optimisation multi-objectif robuste complète, de la conception à la résolution, qui a été composée par les étapes suivantes : a) une petite révision de l'application ; b) la description du problème général ; et c) la formulation, résolution et discussion d'une étude de cas.

En considérant des facteurs comme l'importance de l'application réelle dans l'ingénierie et l'industrie, l'intérêt de la communauté scientifique sur le sujet, et la possibilité de traiter le problème par intervalles, il a été choisi un problème d'optimisation multi-objectif de synthèse des contrôleurs PID. Donc sur le problème général, un cas spécifique a été proposé et résolu en utilisant les algorithmes [I]RMOA I, [I]RMOA II et [I]RMOEA.

### 5.1 - Révision sur les Contrôleurs PID

Les contrôleurs PID, soient analogiques et digitaux, sont utilisés en diverses applications industrielles pour contrôler flux de liquides, température des fours et vitesse de moteurs par exemple. La simplicité de fonctionnement et la performance considérée optimale sont souvent citées comme motifs pour le choix de contrôleurs PID dans l'industrie.

En contrôleurs PID, l'ajustement de variables de contrôle est fait de façon automatique selon les mesures obtenues de la sortie du processus  $z(t)$  comparé à une *valeur de référence*  $r(t)$  dans le temps  $t$ . La différence entre les valeurs  $r(t)$  et  $z(t)$  est le *signal d'erreur*  $e(t)$  :

$$e(t) = r(t) - z(t), \tag{5.1}$$

qu'on désire minimiser ou éliminer, en stabilisant  $z(t)$  au niveau plus proche de  $r(t)$ . L'action de contrôle agit dans le processus de façon proportionnel,

intégrale et dérivée conforme l'expression :

$$m(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt}, \quad (5.2)$$

soient  $m(t)$  la sortie du contrôleur,  $K_p$  le gain proportionnel,  $K_i$  le gain intégrale et  $K_d$  le gain dérivatif. Un schéma simple d'un contrôleur PID associé à un processus générique ou une *planta* a été présenté dans la Fig.5.1.

Le vecteur de paramètres  $\mathbf{K} = \{K_p, K_i, K_d\}$  du PID est ajusté en fonction du système à être régler. On suppose la variable à manipuler prennent les valeurs définies par le contrôleur, guidées par l'erreur existant entre  $r(t)$  et  $e(t)$ . La tâche de sélectionner les valeurs appropriées pour les gains  $\mathbf{K}$  est appelé de *calibration ou régulation de PID*. Il y a différentes méthodes pour le calibrage de ces contrôleurs, lesquels ont été cités sur le texte principal. La méthode employée ici est basée sur l'optimisation.

*Références: Astrom et Hugglund [5], Ziegler-Nichols [129], Cohen-Coon [22], Visioli [122], Wang et Cai [125], et Chidambaram et Sree [19].*

## 5.2 - Description du Problème

L'optimisation de systèmes de contrôle PI, PD ou PID pour un système générique signifie trouver le mieux ensemble de paramètres  $K_p$ ,  $K_i$  et  $K_d$  que minimise quelques métriques/spécifications de performance de la réponse indicielle du système. Les spécifications dans le domaine du temps sont observées dans les systèmes où la réponse est considéré *stable*. La Fig.5.2 présente l'identification de ces métriques via un exemple de réponse indicielle d'un système en boucle fermée.

Le *dépassement*  $M_p$  est la valeur de la réponse indicielle du système au-dessus de  $r(t)$ . Le *temps de montée*  $t_s$  marque la vitesse du système pour répondre à l'entrée. Ils sont valeurs typiques à  $t_s$ : a) de 10% à 90%, b) de 5% à 95% e c) de 0% à 100%. Dans ce travail, l'option (c) a été choisie pour  $t_s$ . De même façon, le *temps de dépassement maximum*  $t_p$  est le temps mesuré dès le début du procédé jusqu'au plus haut dépassement  $M_p$ . Le *temps de réponse*  $t_a$  marque la stabilisation de la réponse du système dans une région stable (tolérance) proche de la valeur  $r(t)$  que se désire obtenir. La tolérance permise est normalement définie en tour de  $\pm 2\%$  à  $\pm 5\%$ . Excepte dans certaines applications, il est désiré que la réponse indicielle soit suffisamment rapide et amortie. Ainsi, les métriques des réponses listées antérieurement, il est désiré que le processus d'optimisation minimise  $M_p$ ,  $t_s$ ,  $t_p$  et  $t_a$ . En addition, il a été choisi une métrique basée en erreur accumulée pour évaluer

la qualité de la réponse du système. Il a été choisi la *métrique ISE* (Integral Square-Error) donné par l'équation :

$$\text{ISE} = \int_0^{\infty} e^2(t) dt. \quad (5.3)$$

Les mesures de performance et de qualité décrites en haut ont été utilisées pour formuler le problème d'optimisation de synthèse de contrôleurs PID. Cependant, avant de décrire le problème, il a besoin d'établir les conditions de stabilité du système. Normalement, la stabilité est une caractéristique fondamentale en systèmes avec rétroaction, soit comme considérée une pré-condition de projet de systèmes de contrôle linéaires.

Le critère RH (Routh-Hurwitz) a été la méthode choisie pour vérifier la stabilité du système. Il utilise seulement les coefficients de l'équation caractéristique (dénominateur de la fonction de Laplace en boucle fermée) pour conclure sur l'existence de stabilité. L'équation caractéristique est donnée par :

$$a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0, \quad a_n \neq 0, \quad (5.4)$$

soient  $s$  la variable de Laplace et les coefficients  $a_n, \dots, a_0$  sont grandeurs réelles. En partant de (5.4) la Table 5.1, dit Table RH, a été construit pour déterminer les conditions RH de stabilité.

Table 5.1: Coeficientes RH.

$s^n$	$a_n$	$a_{n-2}$	$a_{n-4}$	$\dots$
$s^{n-1}$	$a_{n-1}$	$a_{n-3}$	$a_{n-5}$	$\dots$
$s^{n-2}$	$b_1$	$b_2$	$b_3$	$\dots$
$s^{n-3}$	$c_1$	$c_2$	$c_3$	$\dots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Les coefficients  $b_1, b_2, \dots$  sont :

$$b_1 = \frac{a_{n-1}a_{n-2} - a_n a_{n-3}}{a_{n-1}}, \quad b_2 = \frac{a_{n-1}a_{n-4} - a_n a_{n-5}}{a_{n-1}}, \quad \dots \quad (5.5)$$

Et les coefficients  $c_1, c_2, \dots$  sont :

$$c_1 = \frac{b_1 a_{n-3} - b_2 a_{n-1}}{b_1}, \quad c_2 = \frac{b_1 a_{n-5} - b_3 a_{n-1}}{b_1}, \quad \dots \quad (5.6)$$

Le système est déclaré stable si les termes de la première colonne sont tous positifs ou tous négatifs.

Le critère RH peut être étendu quand les coefficients de l'équation caractéristique sont intervalles. Mais, il y a plusieurs occurrences de mêmes coefficients comme se peut voir en (5.5) et (5.6), en résultant une analyse de stabilité très pessimiste. Donc, Kharitonov a développé une technique pour traiter les coefficients intervalles. Avant, de l'introduction de la technique de Kharitonov, considère l'équation caractéristique intervalle :

$$[a_n]s^n + [a_{n-1}]s^{n-1} + \dots + [a_1]s + [a_0], \quad 0 \notin [a_n], \quad (5.7)$$

et maintenant, les polynômes de Kharitonov sont :

$$p_1 : \bar{a}_n s^n + \underline{a}_{n-1} s^{n-1} + \underline{a}_{n-2} s^{n-2} + \bar{a}_{n-3} s^{n-3} + \bar{a}_{n-4} s^{n-4} + \dots \quad (5.8)$$

$$p_2 : \bar{a}_n s^n + \bar{a}_{n-1} s^{n-1} + \underline{a}_{n-2} s^{n-2} + \underline{a}_{n-3} s^{n-3} + \bar{a}_{n-4} s^{n-4} + \dots \quad (5.9)$$

$$p_3 : \underline{a}_n s^n + \bar{a}_{n-1} s^{n-1} + \bar{a}_{n-2} s^{n-2} + \underline{a}_{n-3} s^{n-3} + \underline{a}_{n-4} s^{n-4} + \dots \quad (5.10)$$

$$p_4 : \underline{a}_n s^n + \underline{a}_{n-1} s^{n-1} + \bar{a}_{n-2} s^{n-2} + \bar{a}_{n-3} s^{n-3} + \underline{a}_{n-4} s^{n-4} + \dots \quad (5.11)$$

Le système est déclaré stable si  $p_1$ ,  $p_2$ ,  $p_3$  et  $p_4$  sont stables, selon le critère RH. Pour systèmes d'ordre  $n \leq 5$ , les conditions de stabilité sont plus simples, comme a été proposé par Anderson au-dessous :

$$n = 2 : \text{ stable si } a_2, a_1, a_0 < 0 \text{ ou } a_2, a_1, a_0 > 0 \quad (5.12)$$

$$n = 3 : \text{ stable si } p_1 \text{ est RH} \quad (5.13)$$

$$n = 4 : \text{ stable si } p_1, p_2 \text{ sont RH} \quad (5.14)$$

$$n = 5 : \text{ stable si } p_1, p_2, p_3 \text{ sont RH} \quad (5.15)$$

Une fois que les conditions de stabilité sont satisfaites, le problème de contrôle PID peut être formulé.

En récente recherche sur le travail de contrôle robuste de systèmes intervalles, il a été observé un intérêt grandissant de la communauté scientifique sur le sujet. Il a des études qu'associe le terme "contrôle robuste" à capacité du système à maintenir la stabilité quand la valeur de ses paramètres change. Dans l'optimisation, le but souvent est trouver le ensemble de paramètres que maximise la performance de la réponse indicielle, dans une région garantie robuste. Pour d'autres les méthodes étudiés, un algorithme génétique a été employé pour minimiser l'erreur ISE dans une formulation mono-objectif qui utilisait le pire cas du scénario de la fonction de contrainte. Dans ce cas, les

incertitudes ont été considérées sur les paramètres du système ; les paramètres des contrôleurs ont été fixés. Dans une autre étude, des intervalles ont été utilisés pour trouver les poles du système, et aussi en partant d'équation caractéristique intervalle des paramètres de contrôleurs intervalles ont été obtenus via l'équation Diophantine intervalle. Finalement, le problème PID a été formulé comme un problème multi-objectif, soit le système à être optimisé sous la forme d'intervalles. Les références sur ces travaux de recherche sont présentées à la fin de la section. La formulation du problème robuste choisie dans cette thèse peut être considéré comme un "mélange" de ces études, car: a) le traitement de la stabilité est fait par moyen d'intervalles ; b) les contrôleurs sont écrits en format d'intervalles ; et c) la formulation du problème est multi-objectif. Ainsi, soient le vecteur  $\mathbf{K}$  les gains du système et  $\mathbf{p}$  le paramètre d'incertitude, un problème robuste multi-objectif pour la synthèse de contrôleurs PID d'un système générique de ordre  $n$  peut être écrit comme :

$$\begin{aligned} \min_{\mathbf{K} \in [\mathbf{K}]} \max_{\mathbf{p} \in [\mathbf{P}]} & f_1, f_2, f_3, f_4 \\ \text{s.c :} & g_1 = \text{critère RH ;} \\ & g_2 = \max z(\mathbf{K}, \mathbf{p}, t) < \text{écart acceptable de } u(t), \quad t = t_0 ; \\ & g_3 = \text{singularités (Analyse par Intervalles) ;} \\ & g_4 = \text{autres singularités.} \end{aligned} \tag{5.16}$$

$$\begin{aligned} \text{Avec :} & f_1 = \max z(\mathbf{K}, \mathbf{p}, t), \quad t \in [0, \infty] ; \\ & f_2 = \arg \max_{t \in [0, \infty]} (f_1) ; \\ & f_3 = \min_{t \in [0, \infty]} \{t \mid z(\mathbf{K}, \mathbf{p}, t) = r(t)\} ; \\ & f_4 = \text{ISE.} \end{aligned}$$

Normalement, le dépassement  $M_p$  ( $f_1$ ) et le temps de dépassement maximum  $t_p$  ( $f_2$ ) sont spécifications contradictoires, cependant que le temps monté  $t_s$  ( $f_3$ ) et  $t_p$  ne sont pas. Par fois, il y a des situations que  $M_p$  et  $t_p$  n'existe pas comme dans les systèmes sur-amortis. La métrique ISE ( $f_4$ ) inséré bénéficie les réponses avec mineur erreur accumulée dans période transitoire. En (5.16), la contrainte  $g_1$  est lié à la vérification de la stabilité via Routh-Hurwitz, Kharitonov et Anderson. La contrainte  $g_2$  disqualifie les réponses "mauvaises" en régime stationnaire, définies au temps  $t = t_0$ . Et,  $g_3$  garantit l'utilisation de méthodes intervalles. Par exemple, on considère l'intervalle  $[a, b]$ ,  $a \in \mathbb{R}$  et  $b \in \mathbb{R}$ . Si  $a < 0$  et  $b > 0$ , les calculs  $\sqrt{[a, b]}$  et  $1/[a, b]$  représentent des singularités, car ces opérations ne sont pas définies en Analyse par

Intervalles basée en systèmes finis, qui est le cas cette thèse. Finalement,  $g_4$  traite d'autres singularités. Par exemple,  $g_4$  sauve la formulation de produire résultats impossibles tels que  $t_s < 0$ .

*Références: Ogata [91], Kharitonov [70], Anderson et al. [3], Dahleh et al. [25], Pujara et Roy [96], Henrion et Bachelier [55], Hsu et Yu [59], Lordelo et al. [77], Takahashi et al. [119].*

### 5.3 - Étude de Cas

La formulation proposée en (5.16) peut être employée dans plusieurs problèmes de systèmes de contrôle via PID. Donc, il est nécessaire préciser l'étude de cas abordé ici. Le cas considéré est simple. Cependant, la finalité est de détailler le processus d'optimisation robuste via les algorithmes proposés. Le lecteur intéressé peut étendre la procédure à des problèmes que la fonction de transfert soit de ordre  $n \leq 4$ . La limitation d'ordre se justifie, car il y a besoin de définir la fonction d'inclusion du système de forme analytique. Ainsi, si la réponse du système dépend du calcul des racines du polynôme caractéristique, alors la solution analytique "fermée" seulement existe pour équations de quatrième ordre.

Pour mieux décrire de cette étude de cas, cette section a été fragmentée dans les parts suivantes : a) Caractérisation du system ; b) Réponse Temporelle du Système ; c) Formulation du Problème Multi-Objectif Robuste ; d) Fonctions Objectif ; d) Analyse des Fonctions Objectif et de Contrainte ; et e) Expérimentation et Résultats.

#### 5.3.1 - Caractérisation du Système

Dans l'étude de cas, on considère un système de contrôle linéaire, invariant au temps, sans retard, sans bruit d'entrée et avec le contrôle analogique proportionnel plus dérivatif pour répondre à une entrée échelon unitaire. Le système a été présenté dans la Fig.5.3. La fonction de transfert résultant  $G(s)$  pour le cas est donné par :

$$G(s) = \frac{H(s)C(s)}{1 + H(s)C(s)} = \frac{K_p + (K_d)s}{s^2 + (1 + K_d)s + K_p} . \quad (5.17)$$

#### 5.3.2 - Réponse Temporelle du Système

En appliquant un échelon unitaire ( $r(t) = 0, t = 0; r(t) = 1, t > 0$ ) et réalisant la transformé inverse Laplace est obtenu la réponse du système

$z(K_p, K_d, t)$  suivante :

$$z(K_p, K_d, t) = 1 + e^{-\frac{(1+K_d)t}{2}} \left( -\cosh\left(\frac{\sqrt{\alpha}}{2}t\right) + \frac{(-1 + K_d)}{\sqrt{\alpha}} \sinh\left(\frac{\sqrt{\alpha}}{2}t\right) \right) \quad (5.18)$$

avec :

$$\alpha = (K_d + 1)^2 - 4K_p. \quad (5.19)$$

En considérant l'incertitude  $p$  dans la précision du contrôleur analogique, les gains  $K_p$  et  $K_d$  doivent être réécrits comment  $K_p \pm p$  et  $K_d \pm p$ . L'action des contrôleurs  $K_p$  et  $K_d$  sur le plant est linéaire, cependant l'effet de l'incertitude  $p$  sur  $K_p$  et  $K_d$  n'est pas triviale. Ainsi, en considérant  $p$ , les formulations en (5.18) et (5.19) peuvent être réécrits comme suit :

$$\begin{aligned} z(K_p, K_d, p, t) = & 1 + e^{-\frac{(1+K_d+p)t}{2}} \left( -\cosh\left(\frac{\sqrt{\alpha'}}{2}t\right) \right) + \dots \\ & \dots + e^{-\frac{(1+K_d+p)t}{2}} \left( \frac{(-1 + K_d + p)}{\sqrt{\alpha'}} \sinh\left(\frac{\sqrt{\alpha'}}{2}t\right) \right) \end{aligned} \quad (5.20)$$

avec :

$$\alpha' = (K_d + p + 1)^2 - 4(K_p + p). \quad (5.21)$$

### 5.3.3 - Formulation du Problème Multi-objectif Robuste

En considérant la réponse temporelle du système en (5.20) et (5.21), le problème multi-objectif robuste pour l'étude de cas est défini comme :

$$\begin{aligned}
& \min_{K_p, K_d \in [0, 15]} \max_{p \in [-0.1, 0.1]} f_1, f_2, f_3, f_4 \\
\text{s.c. : :} \quad & g_1 = s^2 + (1 + K_d + p)s + (K_p + p) \text{ soit RH ;} \\
& g_2 = \max z(K_p, K_d, p, t) < 1.3, \quad t = 3s ; \\
& g_3 = \alpha' > 1 ; \\
& g_4 = \frac{-\sqrt{\alpha'+1}-(Kd+p)}{\sqrt{\alpha'+1}-(Kd+p)} > 1, \quad 0 \notin \sqrt{\alpha'} + 1 - (Kd + p). \quad (5.22) \\
\text{Avec :} \quad & f_1 = \max z(K_p, K_d, p, t), \quad t \in [0, \infty] ; \\
& f_2 = \arg \max_{t \in [0, \infty]} (f_1) ; \\
& f_3 = \min_{t \in ]0, \infty]} \{t \mid z(K_p, K_d, p, t) = r(t)\}, \quad r(t) = 1 ; \\
& f_4 = \text{ISE.}
\end{aligned}$$

### 5.3.4 - Fonctions Objectif

Les fonctions objectif  $f_1$  et  $f_2$  traitent du dépassement  $M_p$  et du temps de dépassement maximum  $t_p$  de  $z(K_p, K_d, p, t)$  (5.20). Pour les fonctions continues et dérivables, la dérivée a valeur s'annule sur les points maximums et minimums. Donc,  $M_p$  et  $t_p$  sont calculés grâce à la relation :

$$\frac{dz(K_p, K_d, p, t)}{dt} = 0, \quad (5.23)$$

Il en résulte :

$$t_p = \frac{1}{\sqrt{\alpha'}} \ln \left( \frac{1 - (\sqrt{\alpha'} + (K_d + p))^2}{1 - (\sqrt{\alpha'} - (K_d + p))^2} \right), \quad (5.24)$$

et

$$M_p = z(K_p, K_d, p, t_s) - 1. \quad (5.25)$$

La déduction de les équations (5.24) et (5.25) ont été faite dans l'Appendice A du document complète en considérant un modèle sans le paramètre d'incertitude  $p$  et pour valeurs fixes de  $K_p$  et  $K_d$ .

La fonction objectif  $f_3$ , que traite du temps de monté  $t_s$ , est aussi obtenue de  $z(K_p, K_d, p, t)$ . En considérant  $t_s$  comme le période pour le système aller atteindre la référence ( $r(t) = 1$ ),  $t_s$  est formulé comment :

$$t_s = \frac{1}{\sqrt{\alpha'}} \ln \left( \frac{-\sqrt{\alpha'} + 1 - (K_d + p)}{\sqrt{\alpha'} + 1 - (K_d + p)} \right) \quad (5.26)$$

Le développement de l'équation  $t_s$  a été réalisé dans l'Appendice B, que sans perdre de généralité, ont été fixés  $K_p$  et  $K_d$ , et le paramètre  $p$  n'a pas été considéré.

La fonction  $f_4$ , réfèrent à métrique ISE, a été calculé de l'équation intégrale (5.3). La résolution de l'intégrale, a été faite dans l'Appendice C de la version complète de la thèse, a produit comme résultat :

$$\begin{aligned} \text{ISE} &= -\frac{\left(1 + \frac{1-(K_d+p)}{\sqrt{\alpha'}}\right)^2}{\sqrt{\alpha'} - 1 - (K_d + p)} \dots \\ &\dots -\frac{2\left(1 - \left(\frac{1-(K_d+p)}{\sqrt{\alpha'}}\right)^2\right)}{-1 - (K_d + p)} - \frac{\left(1 - \frac{1-(K_d+p)}{\sqrt{\alpha'}}\right)^2}{-\sqrt{\alpha'} - 1 - (K_d + p)}. \end{aligned} \quad (5.27)$$

Les fonctions objectif d'inclusion pour cette application sont les fonctions d'inclusion naturelles. Les fonctions objectif formulées en (5.24-5.27) contient seulement des opérateurs mathématiques élémentaires, donc les fonctions d'inclusion sont convergentes et monotones. Cependant, la minimalité et la finesse dépendent du nombre d'occurrences des variables et du type d'opérations que les relie. Ce fait justifie les simplifications réalisées dans les appendices.

### 5.3.5 - Analyse des Fonctions Objectif et de Contrainte

Le traitement de contraintes formulé en (5.22) est similaire au procédé présenté à la Sous-section 3.5.1. Dans cette étude de cas, les tests d'inclusion classifient tout l'espace de recherche en "réalisable" ou "de frontière", selon la précision donnée. Aucun espace n'a été classifié comment "non réalisable" car les fonctions de contraintes n'ont pas été construites pour garantir l'instabilité. La Fig.5.4 présente l'espace réalisable de l'étude de cas. La région en gris foncé représente l'espace réalisable ; en gris clair, les inter-

valles de frontières. Détail pour obtention de la figure: a) algorithme SIVIA (2.3.12) ; b) paramètre de précision  $\varepsilon = 0.1$  ; et c) 590 boîtes réalisables.

Antérieurement, il a été dit que la fonction d'inclusion de l'étude de cas est convergente et monotone. Les Fig.5.5, Fig.5.6 et Table 5.3 confirment, à travers d'un exemple, les propriétés listées en haut importantes pour l'utilisation de méthodes intervalles. Dans l'exemple, on note la propriété  $[\mathbf{K}]_2 \subset [\mathbf{K}]_1 \subset [\mathbf{K}]_0 \Rightarrow \mathbf{f}([\mathbf{K}]_2) \subset \mathbf{f}([\mathbf{K}]_1) \subset \mathbf{f}([\mathbf{K}]_0)$ . Si les fonctions ne sont pas convergentes et monotones, alors elles ne seront pas utiles, car il est impossible d'éliminer des régions avec sécurité en garantissant la globalité des méthodes par intervalles, par exemple.

### 5.3.6 - Expérimentation et Résultats

L'expérimentation a consisté à utiliser les méthodes validés, pour résoudre le problème multi-objectif robuste décrit en (5.22). Le problème a quatre objectifs, donc la visualisation graphique a été limitée pour présenter trois objectifs simultanément. Les résultats d'optimisation de (5.22) ont été organisé dans quatre blocs: a) les enveloppes sur la frontière robuste ; b) les ensembles solutions ; c) les frontières robustes ; et d) l'analyse des solutions choisies dans la réponse du système de contrôle.

Les paramètres de l'exécution sont présentés ensuite. Conforme la formulation (5.22), ils ont été employés  $K_p = [0, 15]$ ,  $K_d = [0, 15]$  et  $P = [-0.1, 0.1]$ . Aussi, les paramètres spécifiques de chaque algorithme sont: a)  $\varepsilon_{\mathbf{x}\#} = 0.25$  et  $\varepsilon_{[\mathbf{x}]} = 0.05$  pour [I]RMOA I ; b)  $\varepsilon_{[\mathbf{x}]} = 0.05$  et  $\varepsilon_{[\mathbf{p}]} = 0.025$  pour [I]RMOA II ; et c)  $p_c = 1$ ,  $p_m = 0.0005$ ,  $n_{pop} = 100$ ,  $n_{gen} = 60$ ,  $n_{bit} = 15$ ,  $\varepsilon_{[\mathbf{p}]} = 0.025$  et  $r_{bis}^{max} = 8$  pour le [I]RMOEA. Dans l'expérimentation qu'analyse les enveloppes, nous avons pris  $\varepsilon_{[\mathbf{p}]} = 0.005$  et  $\varepsilon_{[\mathbf{p}]} = 0.001$  pour les méthodes [I]RMOA II et [I]RMOEA.

Dans les premiers tests, le [I]RMOA I a confronté le [I]RMOA II et le [I]RMOEA. Les résultats ont été montrés en Fig.5.7 et Fig.5.8. Dans ces figures, les enveloppes 3D pour  $\mathbf{K}^-$  et  $\mathbf{K}^+$  ont été remplacées par une visualisation plus simplifiée, qu'utilise seulement des points sur le graphique que définissent  $\mathbf{K}^-$  et  $\mathbf{K}^+$ . Ainsi, la comparaison entre les algorithmes est devenu plus claire. L'enveloppe produite par [I]RMOA I a été comparée avec trois cas des autres algorithmes. Les cas de simulation consistent à vérifier les valeurs de précision du paramètre d'incertitude  $\varepsilon_{[\mathbf{p}]}$  que produisent des frontières robustes correctes. Les résultats des les simulations (voir Fig.5.7 et Fig.5.8) sont illustrant la dépendance de la frontière robuste de  $\varepsilon_{[\mathbf{p}]}$ . Cette dépendance est conséquence que les fonctions objectif d'inclusion n'est sont pas minimales. Aussi, à travers de ces tests, le [I]RMOA I se confirme comme outil de validation.

Les ensembles solutions ont été analysés dans la deuxième batterie de tests. Les résultats du [I]RMOA II et du [I]RMOEA ont été présentés en Fig.5.9 et Fig.5.10 respectivement. En observant les résultats, il est possible de conclure: a) l'ensemble solution trouvée se situe en  $K_p = 15$  et  $K_d \in [9.5, 13.5]$  ; b) l'ensemble solution n'est pas dépendent de  $\varepsilon_{[p]}$ , mais l'image de l'ensemble l'est ; et c) l'uniformité de distribution des solutions est satisfaisante.

Les frontières ont été aussi analysées en considérant les mêmes valeurs de  $\varepsilon_{[p]}$  pour contraster les algorithmes [I]RMOA II et [I]RMOEA. Les résultats (voir Fig.5.11) ont montré la cohérence entre les deux méthodes. Une fois que les deux algorithmes sont différents, cette information est suffisante pour valider les deux et aussi les ensembles solutions trouvées.

Le problème d'optimisation multi-objectif proposé en (5.22) ne peut pas être considéré concluant avec seulement l'ensemble solution et son image robuste. Pour compléter la tâche d'optimisation multi-objectif, il est nécessaire prendre la décision sur une solution spécifique choisir, c'est à dire, c'est important déterminer la *solution robuste de projet*  $\mathbf{K}^{**} = \{K_p^{**}, K_d^{**}\}$  qu'optimise l'action du contrôleur. Deux  $\mathbf{K}^{**}$  ont été choisi: un ensemble du [I]RMOA II, et l'autre du [I]RMOEA. Le critère adopté pour la prise de décision a été basé sur une sorte de fonction objectif normalisée. Ainsi, en dénotant l'ensemble solution robuste par  $\mathbf{K}^*$  et par  $f_i^{\max}(\mathbf{K}^*, \mathbf{p})$  le maximum valeur de performance de la fonction objectif  $f_i$  selon  $\mathbf{K}^*$ ,  $\mathbf{K}^{**}$  est formulé comme :

$$\mathbf{K}^{**} = \arg \min_{\mathbf{K}^* \in \mathbf{K}^*} \sum_{i=1}^{n_f} \frac{f_i^{\max}(\mathbf{K}^*, \mathbf{p}) - f_i(\mathbf{k}^*, \mathbf{p})}{f_i^{\max}(\mathbf{K}^*, \mathbf{p})}, \quad \mathbf{p} \in \mathbf{P}, \quad (5.28)$$

soient  $f_i^{\max}(\mathbf{K}^*, \mathbf{p}) > 0$  garanti par (5.24)-(5.27). Ainsi, en considérant la précision  $\varepsilon_{[p]} = 0.001$ , la solution robuste choisi par [I]RMOA II a été  $K_p^{**} = 14.9854$  et  $K_d^{**} = 9.6240$ , et pour le [I]RMOEA a été  $K_p^{**} = 15.0000$  et  $K_d^{**} = 9.6088$ . Selon les métriques de performance du système formulé en (5.24)-(5.27), la solution du [I]RMOA II ( $K_p^{**} = 14.9854$ ,  $K_d^{**} = 9.6240$ ) a produit  $t_s = 0.3411$ ,  $t_p = 0.5730$ ,  $M_p = 0.0305$  et  $ISE = 0.2031$ , et le [I]RMOEA ( $K_p^{**} = 15.0000$ ,  $K_d^{**} = 9.6088$ ) en  $t_s = 0.3409$ ,  $t_p = 0.5728$ ,  $M_p = 0.0307$  et  $ISE = 0.2034$ . La réponse du système en considérant la solution du [I]RMOA II a été montré dans la Fig.5.12. Le résultat du [I]RMOEA est similaire, et donc il a été omis. On observe dans la figure que la réponse indicielle a été réellement optimale, car la réponse a été stable, rapide, presque sans dépassement et avec la réponse en régime dedans la tolérance et le plus important, en prisant compte les incertitudes de la formulation.

## Chapitre 6 - Conclusions et Perspectives

### 6.1 - Conclusions

Le choix d'étudier et proposer des algorithmes pour traiter la philosophie d'optimisation robuste que considère le pire cas est lié à la décision d'utiliser l'Analyse par Intervalles ; le calcul des incertitudes se fait de façon naturelle dans les méthodes par intervalles. En traitant les incertitudes du système d'optimisation comme un paramètre isolé, le calcul d'incertitude peut être encore plus intuitif. Si le paramètre d'incertitude apparait une seule fois dans l'expression formelle de la fonction objectif robuste, donc avec une seule évaluation de la fonction objectif, il est possible de calculer tout l'incertitude avec le minimum pessimisme (voir Sous-section 3.5.1). S'il y a une multiplicité d'occurrences du paramètre d'incertitude, alors il est nécessaire de créer des sous-pavages pour calculer plus précisément la performance robuste d'une solution particulière. Il est important de noter qu'il y a des avantages et des inconvénients de considérer le pire cas. Par exemple, dans l'optimisation de dispositifs, l'image d'une solution robuste peut, souvent, être irréalizable dans l'espace des objectifs. Par contre, il y a des situations de risque tel qu'il soit inadmissible une erreur. Donc, le pire cas est la philosophie indiquée pour ce dernier scénario robuste.

Le problème d'optimisation robuste est plus complexe que l'optimisation traditionnelle, car il considère les définitions  $\mathbf{f}(\mathbf{x}, \mathbf{p}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_f}$ ,  $\mathbf{x} \in \mathbb{R}^{n_x}$  et  $\mathbf{p} \in \mathbb{R}^{n_p}$ . Par conséquent, l'ordre du système augmente par un facteur  $n_x \times n_p$ . Si le problème d'optimisation avait seulement des fonctions convexes, la résolution du problème est simple, rapide et précise à travers l'application de méthodes dirigées par dérivées. Cependant, dans la phase de planification de la thèse il a été défini de développer des méthodes pour l'optimisation robuste indépendante des caractéristiques particulières des fonctions objectif et de contraintes, par exemple non convexité et discontinuité de la frontière robuste.

Les méthodes proposées [I]RMOA I, [I]RMOA II et [I]RMOEA présentent, en cet ordre, l'évolution temporelle de la création des algorithmes dans cette thèse. Chaque un de ces algorithmes représente une contribution. Premièrement, l'abstraction de la frontière Pareto robuste a été fondée avec les résultats du [I]RMOA I, à travers la construction d'enveloppes sur la frontière robuste. Une fois que le [I]RMOA I est calibré pour un scénario robuste spécifique  $\mathbf{P}$ , la méthode est prête à gérer l'autre enveloppe pour un autre scénario robuste différent de  $\mathbf{P}$ . De l'autre côté, le [I]RMOA II recherche des solutions robustes à partir d'une boîte qui contient toute la région d'espace de recherche. Les régions irréalizables sont éliminées pour les fonctions de

contraintes, écrites sur la forme d'intervalles. En utilisant la philosophie pessimisme, les régions réalisables sont éliminées lorsque on a la garantie que cette région ne contient pas de solutions robustes. Ainsi, le point fort du [I]RMOA II est la garantie d'enregistrer tous les points de la frontière, avec une précision, dedans les intervalles non éliminés. Tant [I]RMOA I tant [I]RMOA II sont dépendants des dimensions des variables de projet et le paramètre d'incertitudes, cas les bisections utilisant ces espaces. Par contre, dans le [I]RMOEA la recherche est stochastique et l'influence des dimensions n'augmente pas l'effort calculatoire. Cependant, le [I]RMOEA ne garantie pas trouver les solutions robustes. La contribution des méthodes intervalles est limitée au calcul de l'interférence des incertitudes et aussi dans la technique de niche NB[I]. D'entre autres aspects, une caractéristique vérifiée dans les algorithmes [I]RMOA I et [I]RMOA II, et commun dans les méthodes intervalles, est la convergence du procédé de recherche, c'est à dire, les étapes dans les processus de recherche sont tous déterministes, et les algorithmes finissent de façon globale et naturelle selon les valeurs des paramètres de précision définis. Par contre, dans le [I]RMOEA la convergence est déterminée en fixant un nombre maximum d'itérations, ou à travers d'autre critère empirique qu'évalue la convergence. Finalement, les méthodes [I]RMOA I et [I]RMOA II ont seulement deux paramètres à régler, et le [I]RMOEA en a plusieurs. Cette information est cohérente avec la nature déterministe des algorithmes [I]RMOA I et [I]RMOA II, et stochastique-déterministe de [I]RMOEA. La Table 6.1 résume la comparaison des algorithmes proposés. On note dans la table qu'il y a des items comme "coût" avaient eu une évaluation subjectif. Ces évaluations sont parties d'observations durant la phase de tests.

Table 6.1: Comparaison entre [I]RMOA I, [I]RMOA II et [I]RMOEA.

Attribut	[I]RMOA I	[I]RMOA II	[I]RMOEA
Classification	déterministe	déterministe	hybride
Recherche	globale	globale	globale
But	envelopper $\mathbf{Y}^*$	envelopper $\mathbf{X}^*$	trouver $\mathbf{X}^*$
Qualité $\mathbf{Y}^*$	non applicable	haut	moyen/haut
Convergence	oui	oui	dépend critère
Garanti	dépend $\{[\mathbf{f}], \varepsilon_{\mathbf{x}\#}, \varepsilon_{[\mathbf{x}]}\}$	oui	non
Coût	haut	moyen/haut	bas
Dimension	sensible	sensible	moins sensible
Paramètres	peu	peu	plusieurs

En analysant les expérimentations réalisées pour le [I]RMOEA, il a été

considéré que l'uniformité dans la frontière de Pareto robuste a été obtenue avec succès, et avec coût bas. Le succès de l'uniformité est dû à la technique de niche NB[I]. Aussi, la métrique MB[I] est présentée comme une technique efficace pour mesurer la qualité d'uniformité dans l'ensemble de Pareto (voir les comparaisons à Sous-section 3.7.2).

Les méthodes d'optimisation par intervalles ont besoin de construire des fonctions d'inclusion convergentes, monotones, minimales et fines. Une fois que ces caractéristiques sont satisfaites, les méthodes par intervalles peuvent éliminer avec sécurité les régions où les solutions du système ne sont pas. En fait, les caractéristiques d'être minimales et fines sont importantes principalement pour réduire le pessimisme. D'entre les fonctions tests utilisés, la fonction d'inclusion naturelle à ZDT3 a présenté le pire cas de pessimisme, car il y a des multiplicités de paramètres. Comme conséquence, le [I]RMOA I que dépende d'exhaustifs calculs de fonctions d'inclusion a eu sa performance affecté. Dans l'application réelle de synthèse de contrôleurs, la construction de fonctions d'inclusion a été une tâche plus difficile. Pour des systèmes d'ordre supérieur à quatre, la construction de fonctions d'inclusion c'est plus compliqué, car n'existent pas des solutions analytiques par ces cas.

L'application des algorithmes [I]RMOA I, [I]RMOA II et [I]RMOEA à une application réelle a établi une connexion importante entre toutes les étapes du procédé d'optimisation. Le problème choisi a été la synthèse de contrôleurs PID. D'entre autres questions, il a été détaillé comment le problème pouvait être modéliser et traiter par intervalles, en considérant tous les inconvénients. La stabilité est, normalement, une caractéristique désirable dans les systèmes de boucle fermée. Donc, les critères de stabilité ont été considérés comme contraints. En addition, les autres contraintes sont liées à singularités des méthodes intervalles ont été aussi traités comme contraintes. Le résultat du traitement de contraintes (voir Fig.5.4) démontre l'efficacité des méthodes par intervalles à séparer les régions que sont réalisables de celles qui ne le sont pas. L'étude de cas a été finie avec l'application des algorithmes proposés. Les résultats obtenus ont été comparés, comme les frontières et les ensembles solution. Les résultats ont montré la cohérente entre eux, donc prouvant que les trois méthodes produisent des solutions correctes.

*Références: Boyd et Vandenberghe [13], Jaulin et al.[64], Benhamou et Older [10], et Chabert et Jaulin [17]*

## 6.2 - Perspectives

L'optimisation robuste multi-objectif est un problème de coût haut par définition ( $\mathbf{f}(\mathbf{x}, \mathbf{p}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p}$ ). Donc, la plupart des perspectives consistent

à améliorer le temps d'exécution des algorithmes intervalles. L'utilisation de dérivées en problèmes continus, l'emploi d'architecture parallèle dans quelques procédés, le enregistrement et manipulation de boîtes intervalles en arbres sont exemples de comme des algorithmes pouvant améliorer ses performances. Aussi, il est désiré développer la métrique MB[I] par traiter l'espace multidimensionnel. De plus, la technique de niche NB[I] est facile d'adapter, en principe. Donc, NB[I] sera employé à différents algorithmes évolutionnaires et aussi aux autres algorithmes proposés dans cette thèse.

Parte II

TESE EM PORTUGUÊS

## 1. INTRODUÇÃO

Os métodos de otimização têm sido utilizados como suporte à tomada de decisões em diferentes aplicações reais. Em muitos casos, os problemas de otimização podem ser difíceis de resolver, como por exemplo, quando o modelo matemático do problema for não linear e/ou não convexo, composto por parâmetros multi-dimensionais e com imagem no espaço dos objetivos multi-modal. A tarefa de otimização pode complicar-se quando o contexto for multi-objetivo. Em adição, considere um cenário onde o sistema de otimização está corrompido por vários tipos de incertezas as quais podem prejudicar o real desempenho das soluções supostamente ótimas. O tomador de decisão pode eventualmente valer-se de métodos que não computem incerteza e em seguida encontrar soluções confiáveis acreditando em sua experiência e intuição, ou ainda empregar técnicas de análise de sensibilidade para ajudar em suas escolhas. Alternativamente, os tomadores de decisão podem criar uma versão robusta para as funções objetivo através da inserção de um parâmetro que simule a ação das incertezas. Certamente, a melhor maneira de resolver depende do caso. Para ilustrar o problema multi-objetivo robusto, considere o problema de minimização descrito abaixo

$$\left\{ \begin{array}{l} \min_{\mathbf{x} \in \mathbb{R}^2} \quad f_1(\mathbf{x}) = (x_1 - c_1)^2 + (x_2 - c_2)^2; \\ \quad \quad \quad f_2(\mathbf{x}) = c_3x_1 - c_4x_2. \\ \\ s.a. : \\ \quad \quad \quad g(\mathbf{x}) = x_1 + x_2 \leq 0. \end{array} \right. \quad (1.1)$$

Sendo  $c_1, \dots, c_4$  coeficientes positivos;  $x_1$  e  $x_2$  variáveis reais;  $f_1(x_1, x_2)$  e  $f_2(x_1, x_2)$  as funções objetivo; e  $g(x_1, x_2)$  a única função de restrição de desigualdade. A otimização multi-objetivo, aplicada a problemas como em (1.1), tem como finalidade encontrar um conjunto de pontos solução viáveis tais que: a) haja diversidade entre os pontos; b) nenhum ponto dentro do conjunto solução seja considerado pior que qualquer outro em todos os objetivos; e c) nenhum ponto fora do conjunto seja melhor que qualquer outro dentro do conjunto em todos os objetivos. Neste contexto, suponha que se aplique um algoritmo de otimização sobre um problema multi-objetivo cujas variáveis representem medidas de peças, e as funções, o desempenho do

equipamento. Suponha ainda que se obtenha sucesso nos itens (a), (b) e (c), e com o conjunto de pontos solução disponível, o tomador de decisão defina-se por uma solução que melhor atenda aos interesses da empresa. Assuma também que a confecção das peças seja realizada. Se o equipamento para executar tal tarefa não oferecer a precisão necessária, então um novo grupo de medidas deveria ser encontrado. Certamente, a precisão do torno pode ser embutida na formulação do problema. Mas, isto é desnecessário se for utilizado um algoritmo para otimização robusta que leve em conta a imprecisão de  $\mathbf{x}$ .

Considere o conjunto de pontos solução para o problema (1.1). A escolha do tomador de decisão pode levar em conta a sensibilidade da solução. Por exemplo, imagine que uma pequena alteração do ponto solução escolhido implique em uma indesejável variação nas funções objetivo, ou equivalentemente, permita que haja um eventual ajuste nos valores dos coeficientes para deixar o modelo descrito em (1.1) mais correto. Nestes casos, uma outra solução pode surgir como melhor alternativa simplesmente se ela apresentar menos sensibilidade a presença de ruído na formulação. As técnicas de análise de sensibilidade podem ajudar nas decisões. Por outro lado, se o problema for reformulado do ponto de vista de otimização robusta, a sensibilidade pode ser garantida por definição.

A presença de incertezas pode deixar a expressão (1.1) vulnerável e sujeita a falhas, sendo necessária a aplicação de métodos suplementares para resolução do problema. Entretanto, a introdução das incertezas no sistema de otimização deve ser realizada com cautela, observando-se por exemplo, sua procedência e sua provável intensidade. A correta inserção das incertezas na formulação transforma o problema de otimização tradicional em um problema de otimização robusta, sendo a versão multi-objetivo o objeto de estudo desta tese.

## 1.1 Justificativa

Considere inicialmente  $\mathbf{x} \in \mathbb{R}^{n_x}$  o vetor de variáveis (ou vetor de parâmetros de projeto) e  $\mathbf{f}(\mathbf{x}) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_f}$  o vetor funções objetivo. Um problema de otimização multi-objetivo irrestrito pode ser escrito como:

$$\min_{\mathbf{x} \in \mathbb{R}^{n_x}} \mathbf{f}(\mathbf{x}) \quad (1.2)$$

Agora, assumamos que o sistema de otimização esteja sujeito à presença de incertezas, sejam elas vinculadas aos parâmetros ou às funções objetivo. Certamente, a expressão (1.2) não representa um modelo matemático que con-

temple a interferência das incertezas e portanto, o vetor de funções objetivo em (1.2) deve ser, de alguma forma, alterada.

As fontes de incerteza são diversas. Por exemplo, usualmente os problemas de otimização envolvem cálculo numérico com variáveis reais que não podem ser representadas exatamente por um equipamento computacional, porque há limite de memória para discretizá-las. Em algum ponto, números muito grandes ou muito pequenos são arredondados e com isso há perda da precisão. Além disso, as incertezas estão presentes ao medir e ao construir  $\mathbf{x}$ . Claramente, a formulação em (1.2) não cobre estes casos. Por tudo isso, assumamos  $\mathbf{x}$  impreciso. O parâmetro de projeto atualizado  $\tilde{\mathbf{x}}$ , dependente ou independente de  $\mathbf{x}$ , poderia ser escrito como

$$\tilde{\mathbf{x}} = \mathbf{x}(\epsilon) \text{ ou } \tilde{\mathbf{x}} = \mathbf{x} \pm \epsilon, \quad (1.3)$$

sendo  $\epsilon \in \mathbb{R}$  interpretado como um parâmetro de tolerância, por exemplo.

O ambiente no qual o sistema de otimização está imerso também é fonte de incertezas. Por exemplo, fatores como umidade, temperatura, pressão e vibração podem afetar a avaliação das funções objetivo. Uma alternativa seria acrescentar um novo parâmetro que quantifique esta interferência. Assim, o vetor de funções objetivo poderia ser reformulado como

$$\mathbf{f} = \mathbf{f}(\mathbf{x}, \xi), \quad (1.4)$$

com  $\xi \in \mathbb{R}$  simbolizando o parâmetro dependente da influência do ambiente.

Similarmente ao vetor de variáveis de projeto, as incertezas podem estar presentes na medição de parte dos argumentos de  $\mathbf{f}$ . Assumindo-se o desvio entre os valores medido e real como fixo, uma possível interpretação seria

$$\mathbf{f} = \mathbf{f}(\mathbf{x}, \varepsilon), \quad (1.5)$$

sendo  $\varepsilon \in \mathbb{R}$  o valor da imprecisão na medida de  $\mathbf{f}$ .

Substituindo-se todas as fontes de incerteza acima, representadas pelos parâmetros  $\epsilon$ ,  $\xi$  e  $\varepsilon$ , por único parâmetro  $\mathbf{p}$ , as expressões (1.3), (1.4) e (1.5) podem ser reescritas como

$$\mathbf{f}(\mathbf{x}, \mathbf{p}) = \mathbf{f}(\mathbf{x}, \epsilon, \varepsilon, \xi), \quad \mathbf{p} \in \mathbb{R}^{n_p}. \quad (1.6)$$

Neste trabalho,  $\mathbf{f}(\mathbf{x}, \mathbf{p})$  representa a forma robusta de  $\mathbf{f}(\mathbf{x})$ , ou seja, o vetor de funções objetivo robustas  $\mathbf{f}(\mathbf{x}, \mathbf{p})$  incorpora todas as incertezas identificadas no sistema de otimização.

A discussão em torno de  $\mathbf{f}(\mathbf{x}, \mathbf{p})$  poderia considerar ainda que a intensidade das incertezas variasse no tempo; ou que o projetista, por desconhecimento ou descuido, criasse um modelo falso para o sistema de otimização. No entanto, decidiu-se por limitar a discussão porque a expressão (1.6)

atende aos propósitos deste trabalho. Neste momento, o mais importante é registrar que as incertezas estão, mesmo sendo desprezíveis, sempre presentes no sistema de otimização. Por outro lado, desconsiderar as incertezas em um problema onde elas realmente são relevantes, leva os métodos tradicionais a encontrar soluções ineficientes, o que é indesejado. O interesse em resolver problemas de otimização envolvendo incertezas foi a principal motivação deste trabalho.

## 1.2 Objetivos

A otimização robusta multi-objetivo pode ser considerada como uma linha de pesquisa pouco explorada, mas que recentemente, tem recebido crescente atenção da comunidade científica. Existem diversas questões em discussão que vão desde a interpretação do significado do termo robusto até os algoritmos para resolvê-la. Neste sentido, este trabalho contempla vários aspectos de otimização robusta e busca ser um ponto de partida para outros estudos no mesmo tema.

### 1.2.1 Geral

Considere o parâmetro de projeto  $\mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^{n_x}$  e o parâmetro de incerteza  $\mathbf{p} \in \mathbf{P} \subseteq \mathbb{R}^{n_p}$ . O propósito principal deste trabalho é desenvolver algoritmos que utilizem técnicas intervalares para encontrar o conjunto solução  $\mathbf{X}^* \subseteq \mathbf{X}$  para os problemas de otimização multi-objetivo robusto do tipo

$$\begin{aligned} \min_{\mathbf{x} \in \mathbf{X}} \quad & \max_{\mathbf{p} \in \mathbf{P}} \mathbf{f}(\mathbf{x}, \mathbf{p}) \\ \text{s.a.} \quad & \mathbf{g}(\mathbf{x}, \mathbf{p}) \leq \mathbf{0}, \quad \mathbf{p} \in \mathbf{P}, \end{aligned} \tag{1.7}$$

sendo  $\mathbf{f}(\mathbf{x}, \mathbf{p}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_f}$ , o vetor de funções objetivo; e  $\mathbf{g}(\mathbf{x}, \mathbf{p}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_g}$ , o vetor de funções de restrição de desigualdade.

Três algoritmos intervalares foram desenvolvidos para resolver (1.7), sendo dois deles puramente determinísticos e o terceiro híbrido, usando processos estocásticos para busca de soluções e processos determinísticos para computação das incertezas.

### 1.2.2 Específicos

Concomitantemente ao desenvolvimento de algoritmos para otimização robusta, este trabalho tem como objetivos secundários:

- revisar Otimização Multi-Objetivo, Análise Intervalar e os Algoritmos Evolucionários Multi-Objetivos, pois estes temas são suporte aos algoritmos propostos;

- 
- apresentar outras formulações para otimização robusta, e com isso posicionar a formulação adotada neste trabalho;
  - desenvolver técnicas intervalares para os algoritmos robustos propostos e estender seu uso para outros algoritmos de otimização;
  - propor problemas teste para estudar os algoritmos desenvolvidos, facilitando a comparação entre métodos em outros estudos;
  - validar os algoritmos robustos propostos com problemas teste;
  - analisar a aplicabilidade dos algoritmos robustos em um problema real.

### 1.3 Escopo

Existem diferentes abordagens para investigar e interpretar os problemas de otimização robusta. Conseqüentemente, o desenvolvimento de métodos para resolver problemas desta natureza fica dependente da abordagem escolhida. Por isso, julgou-se importante e necessário apresentar as principais decisões empregadas nesta pesquisa. O escopo deste trabalho se limita a:

- apresentar e discutir os problemas de otimização no formato apenas de minimização - isto facilita a padronização e o entendimento sem perda da generalidade; as expressões para maximização podem ser obtidas de forma análoga às de minimização;
- empregar somente o conjunto dos Números Intervalares Reais - intervalos complexos não serão tratados;
- considerar somente o intervalo como um conjunto onde o valor desejado, que é pontual, está contido - a abordagem de intervalos como uma entidade matemática isolada não será utilizada. No entanto, a diferenciação entre as abordagens sobre intervalos é realizada;
- apresentar apenas conceitos básicos sobre a Otimização Multi-Objetivo, Análise Intervalar e Algoritmos Evolucionários Multi-Objetivo que suportem o desenvolvimento teórico e prático das propostas deste trabalho - referências estão disponibilizadas ao longo do texto para consultas mais aprofundadas;
- utilizar problemas teste para os algoritmos robustos cuja formulação seja composta de funções elementares - desenvolveu-se biblioteca intervalar apenas para funções com operações elementares, ou seja funções que envolvam operações do tipo  $+$ ,  $-$ ,  $\times$ ,  $\div$ ,  $\exp$ ,  $\sin$ ,  $\cos$ , etc;

- desenvolver algoritmos que não necessitem informação do gradiente para realizar o processo de busca - algoritmos intervalares tipo Newton e similares são abundantes. Além disso são limitados a resolver problemas onde o cálculo de derivadas seja possível;
- desenvolver algoritmo evolucionário para resolver o problema robusto definido em (1.7) com estrutura típica de Algoritmos Genéticos - a adaptação de outros algoritmos evolucionários poderá ser realizada por analogia;
- empregar métodos tradicionais de seleção, cruzamento e mutação com parâmetros fixos - testes exaustivos de busca por melhores conjuntos de parâmetros e métodos não serão realizados;
- utilizar cadeia de caracteres binária para representar as variáveis de busca no algoritmo evolucionário proposto - variáveis com codificação real não serão investigadas.

### 1.4 Metodologia

A metodologia científica consistiu principalmente nas fases:

- levantamento do “estado da arte” dos objetos de pesquisa;
- desenvolvimento de biblioteca intervalar;
- elaboração de padrões, definições e do modelo de otimização multi-objetivo robusta;
- desenvolvimento de algoritmos;
- definição de funções teste robustas para validação dos algoritmos;
- aplicação dos algoritmos propostos em um problema real.

A busca por referências bibliográficas sobre otimização robusta multi-objetivo não foi tarefa trivial visto que o tema tem, segundo a pesquisa realizada, limitado acervo bibliográfico. Os trabalhos encontrados foram suficientes para dar suporte aos conceitos de otimização robusta propostos nesta tese. Similarmente, a comunidade científica que estuda e aplica intervalos em algoritmos de otimização multi-objetivo não é numerosa se comparada a seus concorrentes evolucionários. Entretanto, as teorias e algoritmos apresentados nas bibliografias estudadas foram referências importantes para a criação e implementação dos métodos neste trabalho.

Sobre a codificação, há na Web diversas bibliotecas disponíveis gratuitamente que computam a aritmética intervalar. No entanto, decidiu-se por desenvolver a própria, pelos seguintes motivos: a) utilizar padrão próprio de codificação; b) construir código mais portátil para integração com os algoritmos desenvolvidos; e c) liberdade para disponibilizar o código. Quanto à ferramenta computacional, optou-se pelo MATLAB por sua versatilidade, pela sua biblioteca disponível, pelos recursos gráficos e pela facilidade de programação.

Descrever o problema robusto e resolvê-lo via algoritmos intervalares exigiu um formalismo matemático objetivo, coerente e completo para evitar ambigüidade. Assim, primeiramente foi escolhido o modelo matemático para o problema robusto, foco dessa tese, e em seguida foram desenvolvidos os algoritmos para resolver tais problemas. As atividades de criação, correção e adequação de padrões para as nomenclaturas se estenderam durante todo o tempo restante do trabalho.

A implementação de algoritmos para otimização multi-objetivo robusta foi realizada gradativamente. Primeiramente, foram criadas e testadas as bibliotecas intervalares. Em segundo lugar, foi implementado um algoritmo genético multi-objetivo que se tornou a base para o método híbrido proposto para otimização robusta. Na seqüência, os métodos de otimização foram finalmente implementados.

Não foram encontradas funções teste específicas para tratar otimização robusta multi-objetivo conforme a formulação 1.7. Por isso, decidiu-se por criar algumas funções teste para otimização robusta através da adaptação de outras funções de casos não robustos.

A resolução de um problema de engenharia pelos algoritmos propostos teve como meta verificar a aplicabilidade dos métodos. Por isso, julgou-se necessário descrever todo um processo de otimização, contextualizando a concepção teórica do problema, a formulação (funções objetivo e de restrição), o desenvolvimento das funções objetivo, o tratamento das restrições, e finalmente a resolução do problema robusto via os métodos apresentados nessa Tese. Os resultados obtidos foram comparados entre si, demonstrando as características de cada um.

## 1.5 Organização do Trabalho

No Capítulo 2, apresentam-se os números intervalares com suas propriedades, sua origem, definições importantes, semânticas associadas a intervalos, relações e operações aritméticas. O capítulo cobre também algumas questões relacionadas à otimização multi-objetivo, abordando os conceitos, formu-

lações e os métodos para encontrar as soluções Pareto. Ainda, neste capítulo estão apresentadas descrições de alguns algoritmos evolucionários multi-objetivos existentes na literatura, assim como funções teste, métricas e técnicas de nicho para validá-los. O Capítulo 3 trata da descrição completa dos algoritmos robustos propostos. São apresentados os [I]RMOA I e [I]RMOA II que são determinísticos, e o [I]RMOEA que é estocástico-determinístico. O Capítulo 4 valida os métodos robustos via utilização de funções teste robustas. No Capítulo 5, os algoritmos robustos são empregados para resolver um problema real. Finalmente, o Capítulo 6 traz as conclusões.

### 1.6 Considerações Finais

Os problemas de otimização reais são, em grande parte, de natureza multi-objetivo e estão sujeitos a diferentes fontes de incerteza. O projetista do sistema de otimização deve avaliar se as incertezas afetam os resultados. Em caso afirmativo, alguma ação deve ser tomada. Por exemplo, se o projetista definir robustez como a capacidade de uma solução candidata manter o nível de desempenho frente às perturbações originadas pelas incertezas, ele pode agir *a posteriori* utilizando técnicas de análise de sensibilidade e buscar pelas soluções mais estáveis a essas perturbações. Por outro lado, ele pode agir *a priori* inserindo o parâmetro de incerteza na definição do problema e utilizar algoritmos para otimização robusta para encontrar as soluções que resolvem o problema proposto. Considerou-se a segunda opção mais interessante para este trabalho.

Existem diferentes definições para problemas de otimização robusta multi-objetivo. Aqui, adotou-se que o problema robusto deve ser equacionado de forma a permitir que os algoritmos busquem as melhores soluções, segundo a avaliação dos objetivos, quando observado a pior interferência do parâmetro de incerteza, conforme descrito em (1.7).

Os algoritmos [I]RMOA I, [I]RMOA II, [I]RMOEA para resolver o problema robusto proposto em (1.7) constituem o foco principal desta Tese. Observa-se que todos algoritmos utilizam métodos intervalares. Os dois primeiros são puramente intervalares, o último é algoritmo híbrido desenvolvido a partir de um algoritmo genético multi-objetivo. No entanto, nos capítulos seguintes pode ser observado que este trabalho também contribui através do desenvolvimento dos métodos auxiliares aos algoritmos principais, da discussão sobre as diversas formas de se interpretar a otimização robusta, da forma de se tratar as restrições, por exemplo.

## 2. REVISÃO BIBLIOGRÁFICA

A decisão de formular e resolver o problema de otimização robusta conforme proposto no Capítulo 1 envolve duas grandes linhas de pesquisa: a) os Algoritmos Evolucionários Multi-Objetivo; e b) a Análise Intervalar. Por isso, este capítulo foi dedicado para introduzir padrões de nomenclatura, apresentar definições e conceitos de cada uma destas linhas. Entretanto, por razões didáticas, decidiu-se por extrair do item (a) algumas questões básicas sobre a Otimização Multi-Objetivo deixando a seção dos algoritmos evolucionários mais voltada para descrição do processo evolutivo. A seção endereçada à Análise Intervalar cobre questões fundamentais, como por exemplo há uma parte destinada ao histórico da computação intervalar. Justifica-se esta decisão porque o emprego de métodos intervalares é recente e trata-se de uma técnica ainda não tão difundida na comunidade científica, principalmente no Brasil.

### 2.1 *Otimização Multi-Objetivo*

Em engenharia, otimizar significa buscar pelo melhor conjunto de parâmetros que modelam um problema matemático, que é definido em termos de ganho em desempenho e/ou diminuição de custos nos objetivos ou critérios propostos como quesitos de avaliação. Os critérios estão organizados em funções objetivo e a sua avaliação pode ser obtida diretamente da imagem destas funções. Entretanto, nos problemas de otimização multi-objetivo a imagem isolada de um critério não permite afirmar que uma solução é a melhor dentro de um grupo de soluções. Geralmente, todos os objetivos devem ser considerados durante o processo de otimização e o resultado é um conjunto de soluções onde não há uma hierarquia entre elas, a menos que seja declarada alguma regra de preferência. A escolha dos métodos de otimização considera o momento em que é feita esta preferência. Elas podem ser inseridas antes, durante e após o processo de otimização conforme apresentado Marler e Arora [80] e Andersson [4]. Alternativamente, pode-se optar por não informar a preferência por critérios. Portanto, a escolha do método para resolver dado problema multi-objetivo depende da decisão de como será feita a preferência sobre critérios.

Esta seção aborda aspectos fundamentais da otimização multi-objetivo que são importantes no decorrer do trabalho. Mais detalhes e informações sobre o tema multi-objetivo podem ser obtidos em Sawaragi *et al.* [102], Miettinen [84] e Yu [127]. No entanto, nenhuma destas referências trata os problemas multi-objetivos sujeitos à presença de incertezas.

### 2.1.1 Fundamentos e Definições

Usualmente, as variáveis de busca ou de projeto são os parâmetros que se deseja otimizar. Entre outras classificações, elas podem ser: a) *independentes*; b) *dependentes*; e c) *de ambiente*. As variáveis de projeto são independentes quando isoladamente caracterizam algum aspecto do projeto. Dependentes, quando a variável está vinculada a outro parâmetro, ou seja, seu desempenho modifica-se com a alteração de outra variável. Finalmente, as variáveis de projeto podem ser classificadas como variáveis de ambiente. Elas não fazem parte do sistema de otimização diretamente, mas podem interferir nos resultados. Por exemplo, num dado problema a temperatura pode alterar variáveis independentes e dependentes. Andersson [4] ainda classifica as variáveis de projeto como de *estado* e de *operação*. Neste trabalho, as variáveis de projeto independentes, dependentes ou de ambiente são freqüentemente notadas por  $\mathbf{x}$ , e definidas no espaço de projeto ou espaço de busca  $\mathbf{X}$ ,

$$\mathbf{X} \subseteq \mathbb{R}^{n_x}, \quad (2.1)$$

sendo  $n_x$  o número de variáveis de projeto. Adiante, a relação de continência (2.1) reaparece toda vez que for julgado que isso deixará o texto mais claro.

As funções objetivo são expressões matemáticas para avaliar ou medir os critérios estabelecidos pelo projetista utilizando-se como argumentos as variáveis de projeto. Considerando-se um problema multi-objetivo com  $n_f$  critérios, as funções são definidas como  $\mathbf{f}(\mathbf{x}) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_f}$ . No entanto, o vetor de funções objetivo também pode representar uma notação de conjunto, como

$$\mathbf{f}(\mathbf{X}) = \{\mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X}\}. \quad (2.2)$$

A comparação entre vetores tem papel essencial na otimização multi-objetivo. Uma solução só é considerada melhor que outra depois de comparadas as avaliações de todas as funções objetivo do problema. Os operadores necessários para fazer esta comparação são descritos abaixo. Considere  $\mathbf{x}_1, \mathbf{x}_2 \in \mathbf{X}$  e o vetor de funções objetivo  $\mathbf{f}(\mathbf{x}) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_f}$ . O operador de comparação  $\preceq$  é definido como

$$\mathbf{f}(\mathbf{x}_1) \preceq \mathbf{f}(\mathbf{x}_2), \text{ se } \mathbf{f}(\mathbf{x}_1) \leq \mathbf{f}(\mathbf{x}_2), \mathbf{f}(\mathbf{x}_1) \neq \mathbf{f}(\mathbf{x}_2). \quad (2.3)$$

Por analogia, define-se  $\succeq$  e os casos estritos  $\prec$  e  $\succ$ . Adicionalmente,  $\not\prec$  significa não dominância do primeiro para o segundo operador. No contexto de minimização, a expressão  $\mathbf{f}(\mathbf{x}_1) \preceq \mathbf{f}(\mathbf{x}_2)$  é lida como  $\mathbf{f}(\mathbf{x}_1)$  domina  $\mathbf{f}(\mathbf{x}_2)$ ,  $\mathbf{f}(\mathbf{x}_2)$  é dominado por  $\mathbf{f}(\mathbf{x}_1)$  ou ainda,  $\mathbf{x}_1$  é eficiente em relação a  $\mathbf{x}_2$ . O uso dos termos “eficiência” e “dominância” pode gerar confusão dependendo do contexto. Aqui, emprega-se o conceito de ponto eficiente para tratar o vetor de variáveis  $\mathbf{x}$  no espaço das variáveis e o de dominância está voltado para o vetor das funções  $\mathbf{f}(\mathbf{x})$  no espaço das funções.

Os objetivos de um problema real são normalmente diversos em sua natureza e em seus valores quantitativos. Quando o uso de alguma ponderação nas funções objetivo não é realizada ou nenhum agrupamento de critérios é feito, usualmente utiliza-se métodos de comparação aos pares para separar as soluções eficientes. As soluções eficientes são definidas abaixo.

**Definição 1** Considere  $\mathbf{x}^* \in \mathbf{X}$  e  $\mathbf{f}(\mathbf{x}) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_f}$ . A solução  $\mathbf{x}^*$  é ponto eficiente ou Pareto ótimo, se  $\nexists \mathbf{x} \in \mathbf{X}$ ,  $\mathbf{f}(\mathbf{x}) \preceq \mathbf{f}(\mathbf{x}^*)$ . O conjunto de pontos eficientes é simbolizado por  $\mathbf{X}^*$ .

### 2.1.2 Problema de Otimização Multi-Objetivo

Considere  $\mathbf{X} \subseteq \mathbb{R}^{n_x}$  o espaço de busca e  $\mathbf{x}$  a variável de projeto, tal que  $\mathbf{x} \in \mathbf{X}$ . Se  $\mathbf{f}(\mathbf{x}) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_f}$  representa o vetor das funções objetivo e  $\mathbf{g}(\mathbf{x}) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_g}$  expressa as restrições sobre as variáveis de projeto, então, o problema de otimização multi-objetivo pode ser escrito como

$$\begin{aligned} \min_{\mathbf{x} \in \mathbf{X}} \quad & \mathbf{f}(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{x}) \leq \mathbf{0}. \end{aligned} \quad (2.4)$$

O vetor de funções de restrições define sobre  $\mathbf{X}$  o espaço viável  $\mathbf{X}'$ ,

$$\mathbf{X}' = \{\mathbf{x} \in \mathbf{X} \mid \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\}. \quad (2.5)$$

Resolver (2.4), significa encontrar um conjunto  $\mathbf{X}^*$ ,

$$\mathbf{X}^* = \{\mathbf{x}^* \in \mathbf{X}' \mid \nexists \mathbf{x} \in \mathbf{X}', \mathbf{f}(\mathbf{x}) \preceq \mathbf{f}(\mathbf{x}^*)\}, \quad (2.6)$$

sendo  $\mathbf{x}^*$  o ponto Pareto ótimo, como anteriormente. Como consequência da Def. 1, a *fronteira de Pareto*  $\mathbf{Y}^*$  é definida como

$$\mathbf{Y}^* = \mathbf{f}(\mathbf{X}^*). \quad (2.7)$$

A Fig. 2.1 apresenta graficamente um problema multi-objetivo de forma geral. O lado direito ilustra o espaço dos objetivos, e o lado esquerdo ilustra o espaço

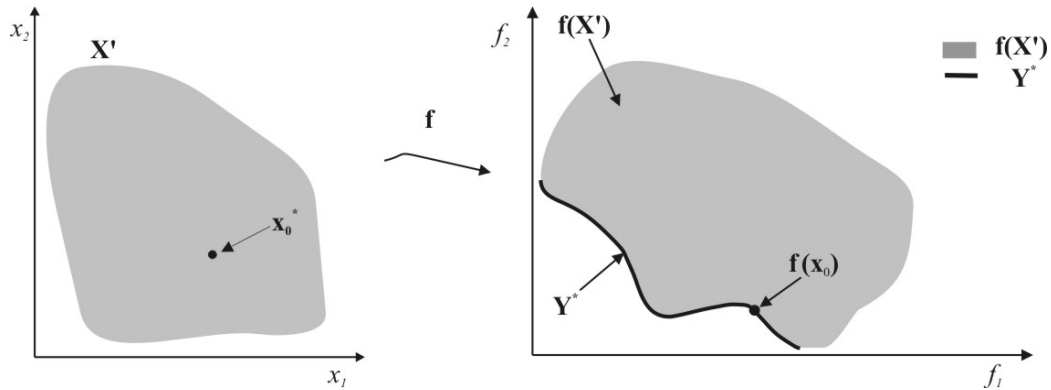


Fig. 2.1: Os espaços de busca e dos objetivos em um problema de otimização multi-objetivo convencional: o espaço viável  $\mathbf{X}'$  e um minimizador  $\mathbf{x}_0^*$ ; a fronteira de Pareto  $\mathbf{Y}^*$ ; as imagens de  $\mathbf{X}'$  e  $\mathbf{x}_0^*$ .

de busca. Ainda, um minimizador  $\mathbf{x}_0^*$  estabelece relação entre os espaços de busca e dos objetivos.

Algumas formulações multi-objetivo, por exemplo em Osyczka [92], e Koski e Silvenoinen [74], utilizam a informação da imagem da solução ideal para minimização  $\mathbf{u}^{\min} \in \mathbb{R}^{n_f}$  para criar uma *função de transformação* sobre  $\mathbf{f}(\mathbf{x})$ . Formalmente,  $\mathbf{u}^{\min}$  é expressado por

$$u_i^{\min} = \min_{i=1, \dots, n_f} f_i(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbf{X}. \quad (2.8)$$

Importante observar que,  $\mathbf{u}^{\min} \notin \mathbf{f}(\mathbf{X})$  quando os critérios são conflitantes. Na prática,  $\mathbf{u}^{\min}$  é uma estimativa baseada nos pontos  $\mathbf{x}$  disponíveis. Como o próprio nome diz, a solução ideal seria a única que, indiscutivelmente, otimizaria todos os objetivos. No entanto, ela raramente existe e, portanto alguma outra solução que esteja próxima da ideal pode vir a atender satisfatoriamente aos requisitos. Esta solução é denominada *solução de compromisso*. A proximidade até a solução ideal pode ser medida, por exemplo, através da aplicação da norma Euclidiana.

### 2.1.3 Classificação dos Métodos Multi-Objetivos

Em problemas multi-objetivo, segundo o ponto de vista do projetista e/ou do tomador de decisão, alguns critérios podem ser considerados mais importantes que outros. Se isso ocorre, o projetista pode adicionar “pesos” a alguns termos nas funções objetivo para realçar essa preferência. Alternativamente, pode-se resolver o problema multi-objetivo em um primeiro momento e em seguida escolher dentre as soluções ótimas uma solução que represente alguma

preferência extra do tomador de decisão. Marler e Arora [80] e Anderson [4] classificaram os métodos multi-objetivos de acordo com a ordem da articulação das preferências no processo de otimização e tomada de decisão. Se o método multi-objetivo realizar alguma manipulação dos objetivos antes da sua execução, ele será classificado como técnica de preferências *a priori*. Por outro lado, ele participará do grupo de métodos de preferências *a posteriori* quando a escolha por soluções acontece após a busca pelos pontos ótimos. Há ainda métodos com apresentações de preferências *progressiva* e *sem articulações* de preferência. Detalhes e mais informações sobre métodos de tomada de decisão são encontrados em Parreiras e Vasconcelos [95] e Parreiras *et al.* [94].

A característica principal dos métodos *a priori* é a articulação das preferências pelos critérios antes de aplicar os procedimentos para encontrar as soluções Pareto. Métodos classificados nesta categoria, utilizam geralmente coeficientes, expoentes e outros tipos de parâmetros para enfatizar quais objetivos são mais importantes naquele problema em questão. O papel do tomador de decisão é fundamental para a escolha destes valores, que muitas vezes, são definidos de forma empírica e/ou baseados em resultados de experiências anteriores. Os parâmetros conectam as funções objetivo, criando uma nova função a ser otimizada chamada de *função de utilidade*. Como exemplos de métodos desta classe citam-se: a) Soma Ponderada Zionts [130]; b) Critério Global Ponderado Yu [126] e Zeleny [128]; e c) método Lexográfico Waltz [124].

Os métodos da categoria de preferências progressivas tem sido aplicados com frequência na área de pesquisa operacional. O ajuste das preferências é feito de forma dinâmica ao longo da busca pelas soluções eficientes. O primeiro passo é conhecer uma solução pertencente à fronteira de Pareto. Em seguida, utilizam-se métodos para determinação de direções de busca para encontrar outros pontos. Como não há necessidade de se fixar as preferências no início do processo, o tomador de decisão pode fazer ajustes locais de preferências durante o processo, diminuindo as chances de cometer erros. Alguns exemplos: a) método da Relaxação dos Objetivos, Takahashi *et al.* [119]; b) método STEM, Benayoun *et al.* [9]; e c) método Steuer, Steuer e Choo [116].

Muitas vezes o projetista está em dúvida sobre como manipular os parâmetros sobre os objetivos antes de iniciar o processo de otimização. Nestes casos, ele pode utilizar-se de métodos para descobrir, dentre as soluções encontradas, quais são Pareto. Por isso, métodos de otimização deste grupo têm o papel principal de elencar várias soluções ao longo de toda a fronteira de Pareto. Algoritmos evolucionários, principalmente os Algoritmos Genéticos, têm sido muito utilizados para soluções *a posteriori*. Esta classe de métodos

tem se mostrado muito eficiente para encontrar pontos sobre a fronteira de Pareto. Detalhes destes algoritmos são apresentados no próximo capítulo. Dentre os algoritmos não evolucionários utilizados *a posteriori* citam-se: a) Programação Física, Messac *et al.* [83]; b) método da Restrição Normal, Messac *et al.* [82]; e c) método de Intersecção Normal à Fronteira, Das e Dennis [28]. As referências sobre os algoritmos evolucionários é apresentada na próxima seção, dedicada somente a esta classe de métodos.

Alguns métodos foram desenvolvidos para permitir que as funções objetivo sejam avaliadas sem interferência direta de pesos. Neste modelo, um método simples sem articulação das preferências tem como função a ser minimizada a soma direta dos objetivos. Como a ordem de magnitude das funções objetivo podem ser diversas, os métodos a seguir prometem suavizar essa distância numérica entre elas. Estes métodos têm sido pouco utilizados e não foram detalhados aqui. Assim, para os leitores interessados em aprofundar nestes métodos, os seguintes trabalhos são sugeridos: a) Nash Arbitrário, Davis [30]; e b) método Rao, Rao e Freiheit [97].

Algumas metodologias multi-objetivo contêm versões em grupos diferentes de manipulação das preferências. Por exemplo, o método da Programação Física foi inicialmente empregado *a priori* em Messac [81] e a formulação utilizada para a Soma Ponderada também é usada em algoritmos *a posteriori*. Importante observar que os algoritmos propostos neste trabalho se enquadram na categoria *a posteriori*.

## 2.2 Algoritmos Evolucionários Multi-Objetivos

Os sistemas naturais são assuntos multi-disciplinares e recentemente várias analogias em sistemas artificiais têm sido realizadas. Por exemplo, existem contextos em que etapas do processo evolutivo foram redefinidas, adequadas e modeladas à otimização. Otimização baseada em processos de formação de colônias de formigas (Dorigo e Stützle [37]), do vôo de pássaros (Kennedy e Eberhart [69]) e da caça de predadores (Laumanns *et al.* [76]) são exemplos. Em cada modelo artificial os termos são importados da biologia e adaptados ao ambiente computacional. Uma vez que todas estas analogias dependem, entre outros detalhes, de uma estrutura de dados, da codificação e computação destes dados para simular os processos naturais, elas estão sujeitas à tecnologia vigente. Talvez por este motivo, os métodos de otimização baseados em mecanismos naturais se tornaram realidade somente nas últimas décadas.

Grande parte das primeiras técnicas de otimização utilizam informações do gradiente. Nestes modelos, freqüentemente chamados métodos deter-

minísticos, o processo de pesquisa pelos pontos solução é pré-determinado, a busca é local e, para encontrar novas soluções, outros pontos de partida devem ser lançados. Em algumas técnicas, detalhes como descontinuidade deixavam o modelo sem funcionalidade. Por outro lado, os métodos estocásticos utilizam procedimentos probabilísticos para guiar a pesquisa para a região de otimalidade. Dentro do grupo estocástico, destacou-se uma classe de algoritmos com as seguintes características: a) eficiência em encontrar as regiões ótimas; b) flexibilidade para adaptar em diferentes problemas; c) independentes do uso de derivadas; e d) permite encontrar múltiplos pontos ótimos em uma única execução. Estes e outros itens caracterizam os Algoritmos Evolucionários que ainda são associados as seguintes terminologias: Computação Evolucionária, Estratégias Evolucionárias, Programação Genética e Vida Artificial.

Algoritmos Evolucionários têm sido cada vez mais aplicados para resolver problemas multi-objetivos. Dentre inúmeros trabalhos científicos dedicados a tratar deste assunto, dois têm sido citados com frequência: Deb em [32], em 2001; e a tese de doutorado de Knowles [72], em 2002. Nestas referências, há uma excelente diversidade de conceitos, comparações e reflexões sobre diversos processos de algoritmos evolucionários multi-critérios. Alguns tópicos dos trabalhos citados foram motivadores para realização dos estudos deste capítulo por se identificarem como possíveis investigações desta Tese.

### 2.2.1 Estrutura Geral dos Algoritmos Evolucionários

Vários métodos de otimização tentam reproduzir artificialmente algum processo natural. O fator motivador para realizar a analogia é que, se determinado mecanismo está presente e é eficaz na natureza, então há expectativa para seu sucesso quando aplicado em modelos artificiais. O custo computacional muitas vezes determina a viabilidade do método. Em especial, grande parte dos Algoritmos Evolucionários são baseados no processo evolutivo das espécies descrito por Darwin. No entanto, o termo “evolucionários” é mais amplo e pode ser aplicado aos métodos que desenvolvem soluções a partir da melhoria das soluções da iteração anterior. O mais importante é que os Algoritmos Evolucionários são similares no propósito de desenvolver soluções de forma evolutiva. De forma geral, os Algoritmos Evolucionários têm estrutura algorítmica apresentada na Tab.2.1. Os parâmetros de entrada variam de método para método, mas estão sempre relacionados a detalhes de formação da população e às probabilidades dos operadores evolucionários. Os passos anteriores ao laço evolutivo, passos 1–2, são destinados ao lançamento da população de soluções candidatas e sua avaliação pelas funções objetivo. No laço evolutivo, passos 3 – 8, os processos de seleção, recombinação e de

Tab. 2.1: Estrutura típica de algoritmos evolucionários.

	<b>entrada</b> ( <i>parâmetros</i> )
	<b>saída</b> ( <i>soluções candidatas</i> )
1	inicialize as <i>soluções candidatas</i> com <i>parâmetros</i> .
2	compute o desempenho das <i>soluções candidatas</i> .
3	se critério de parada é atingido siga para 9.
4	aplique método de <i>seleção</i> sobre as <i>soluções candidatas</i> .
5	aplique operador de <i>recombinação</i> sobre o resultado da <i>seleção</i> .
6	aplique operador de <i>diversidade</i> sobre o resultado da <i>recombinação</i> .
7	compute o desempenho do novo grupo de <i>soluções candidatas</i> .
8	siga para 4.
9	retorne as <i>soluções candidatas</i> .

diversidade agem sobre a população até que algum critério de convergência seja atingido. No passo 9 a população corrente contém a solução que deverá ser escolhida pelo decisor.

Os GAs (Algoritmos Genéticos) constituem a classe de algoritmos evolucionários mais popular. Um modelo básico de Algoritmos Genéticos é mostrado na Tab.2.2. A analogia ao processo evolutivo das espécies inicia-se nos passos 1 e 2, com inicialização do contador de gerações e com a criação das primeiras soluções candidatas. Cada solução é chamada de *indivíduo*, e seu grupo, de *população*. No passo 3, os indivíduos recebem uma nota de desempenho obtida da função objetivo que simula um *ambiente de sobrevivência*. O ciclo evolutivo, marcado pelos passos 4 – 10, prossegue com a *seleção* de indivíduos para gerar a próxima população. Os métodos de seleção tentam preservar, durante as gerações, os indivíduos considerados mais aptos para a evolução. Os indivíduos selecionados participam do processo de propagação de informação genética *cruzamento*, segundo uma dada probabilidade. Sobre o resultado do cruzamento, aplica-se o operador de diversidade e finaliza-se a geração com a nova população. Os passos anteriores se repetem até que algum critério seja satisfeito. Termos como população, cruzamento e mutação variam de algoritmo para algoritmo. No entanto, o mais importante é que esta estrutura atende aos passos principais de um algoritmo evolucionário.

O primeiro GA foi apresentado por Holland [57] em 1975, porém a técnica só foi popularizada por Goldberg [45] em 1989. Goldberg apresentou o SGA (Simple Genetic Algorithm). Neste algoritmo a população tem tamanho

Tab. 2.2: Estrutura típica de Algoritmos Genéticos.

	<b>entrada</b> ( <i>parâmetros</i> )
	<b>saída</b> ( <i>população</i> )
1	inicialize $t$ com 0.
2	inicialize a <i>população</i> com <i>parâmetros</i> .
3	compute o desempenho <i>população</i> .
4	se critério de parada é atingido siga para 11.
5	incremente $t$ .
6	aplique método de <i>seleção</i> .
7	aplique método de <i>cruzamento</i> .
8	aplique método de <i>mutação</i> .
9	compute o desempenho da nova <i>população</i> .
10	siga para 4.
11	retorne a <i>população</i> corrente.

fixo e os indivíduos são codificados por cadeia de caracteres binária. As probabilidades dos operadores de cruzamento e mutação são definidas previamente e passada por meio dos parâmetros de entrada. Após a etapa de definições, o mérito de cada indivíduo é avaliado. Um processo de seleção, chamado pelo autor de *Roleta*, é acionado. Neste método, a probabilidade de escolha de cada indivíduo é proporcional ao desempenho dele na população. O cruzamento é realizado aos pares e, a troca de material genético é feita a partir de um corte na cadeia binária. Os novos indivíduos substituem os anteriores e ficam sujeitos à mutação de seus bits. Termina-se a geração. O algoritmo prossegue ciclicamente a partir dessa nova população e só termina quando algum critério de convergência é alcançado, como por exemplo quando um número máximo de gerações  $n_{gen}$  seja atingido. Observe que os passos citados acima estão de acordo com o algoritmo da Tab.2.2. Durante muito tempo o SGA foi padrão de comparação para outros GAs. Depois dele surgiram outros comprovadamente superiores. Uma referência que aborda muitos aspectos topológicos de GAs voltado exclusivamente para otimização mono-objetivo, foi desenvolvida por Soares em [109]. Mesmo que o interesse nesta Tese seja a estrutura multi-objetiva, os GAs mono-objetivo contêm passos fundamentais para o entendimento dos mecanismos evolucionários.

### 2.2.2 Algoritmos Evolucionários para Problemas Multi-Objetivo

Existem diversos motivos para empregar EAs (Evolutionary Algorithms) para tratar problemas multi-objetivos. Dentre eles, ressalta-se a vantagem de se trabalhar com populações, onde o processo de busca é feito segundo informações obtidas por um grupo de pontos, promovendo a possibilidade de se listar mais soluções Pareto em uma única execução. Estes detalhes deixam os EAs mais versáteis e eficientes na resolução de problemas multi-objetivos. Para a classe de problemas multi-objetivos, os EAs são mais conhecidos por MOEAs (Multi-objective Evolutionary Algorithms). Segundo Gao *et al.* [44], os algoritmos evolucionários podem ser divididos em várias categorias, tais como: a) abordagens por planos de agregação; b) abordagens por população não Pareto; c) abordagens por Pareto; e d) Abordagens por Técnicas de Indução a Nicho. O texto a seguir descreve cada uma destas estratégias.

Os algoritmos que empregam abordagens por planos de agregação têm a finalidade de resolver o problema multi-objetivo do ponto de vista mono-objetivo. Assim, esses algoritmos realizam o tratamento das preferências *a priori*, ponderando as funções objetivo e agrupando-as em apenas uma. Se for permitida a oscilação de pesos, o número de soluções encontradas diferentes tende a aumentar. Entretanto, há casos em que a variação dos pesos produz os mesmos resultados. Agregar objetivos parece natural mas fragiliza a busca por vários pontos não dominados. Por exemplo, quando a fronteira de Pareto é não convexa, os pontos sobre esta região podem não ser encontrados. No entanto, esta agregação fornece resultados satisfatórios se o responsável pela execução conhecer o comportamento das funções objetivo.

Os métodos por abordagem não Pareto, que utilizam subpopulações, sendo cada uma encarregada de uma função objetivo. Um exemplo da abordagem por população não Pareto é o VEGA (Vector Evaluated Genetic Algorithm) apresentado em 1985 por Schaffer [103]. Este método realiza a operação de seleção para cada objetivo separadamente. A população é dividida igualmente de acordo com o número de objetivos. Os operadores genéticos são executados como de costume. O resultado final oferece freqüentemente pontos não dominados encontrados paralelamente em cada uma das subpopulações. O VEGA tem dificuldades para mapear todos os pontos da fronteira quando o espaço das funções é não convexo.

Diferentemente do VEGA, o MOGA (Multi-Objective Genetic Algorithm), apresentado pela primeira vez por Fosenca e Fleming [42], faz uso do princípio da não dominância sobre as soluções. A idéia é conduzir a busca para a fronteira de Pareto ponderando aleatoriamente os objetivos cada vez que os indivíduos são selecionados para o cruzamento. O MOGA de Murata e Ishibuchi [89] distingue-se por separar numa população externa, a cada gera-

ção, os indivíduos eficientes. Após o ciclo evolutivo, as soluções da população externa são comparadas, sendo selecionadas aquelas que são eficientes dentro deste grupo. O MOGA de Murata é apresentado na Tab.2.3.

Tab. 2.3: MOGA - Adaptado de Morata *et al.* [89]

<b>entrada</b> ( <i>parâmetros</i> )
<b>saída</b> ( <i>população</i> )
1 inicialize $t$ com 0.
2 inicialize a <i>população</i> com <i>parâmetros</i> .
3 compute o desempenho <i>população</i> .
4 se critério de parada é atingido siga para 14.
5     se $t > 1$
6         atualize <i>população externa</i> .
7         aplique <i>estratégia elitista</i> .
8     incremente $t$ .
9     aplique método de <i>seleção</i> .
10    aplique método de <i>cruzamento</i> .
11    aplique método de <i>mutação</i> .
12    compute o desempenho da nova <i>população</i> .
13    siga para 4.
14 seleção de <i>população</i> dentro de <i>população externa</i> pelo usuário.
14 retorne a <i>população</i> selecionada.

GAs bem interessantes são aqueles que utilizam técnicas de nicho. Em 1994, Horn *et al.* em [58] publicaram o primeiro trabalho sobre formação de nichos na área multi-objetivo, sendo o primeiro estudo na área mono-objetivo atribuído a Goldberg e Richardson [46] em 1987. A meta principal da formação de nichos é manter a diversidade e com isso elencar pontos distintos sobre a fronteira eficiente. Em poucas palavras, o processo inicia-se encontrando a “vizinhança” entre indivíduos, pois elas indicariam a distribuição da população no espaço. Vizinhos próximos formam uma *subpopulação* que é a chave da diversidade. Em um segundo momento, as técnicas de nicho tentam manter as subpopulações através de funções de *partilha*, cuja finalidade é degradar a função de mérito proporcional ao número de indivíduos da vizinhança. Após esta etapa, faz-se a seleção, cruzamento, etc, ou seja, o algoritmo continua normalmente. Vale a pena registrar que a vizinhança é definida como alguma medida de distância entre dois indivíduos, fato que provoca crescimento considerável no custo de processamento do algoritmo.

### 2.2.3 Diversidade de Algoritmos Evolucionários

Existe uma imensa variedade de algoritmos evolucionários multi-objetivo propostos na literatura. Qualquer um que partindo da estrutura da Tab.2.3, alterando o operador de recombinação e fixando todos os outros procedimentos, valida o algoritmo transformado como um novo método. Seguindo este raciocínio, pode-se modificar o tipo de mutação, seleção, estratégia de recolocação da população, etc. Se for realizado um cálculo combinatorial, contabilizando todas as variações propostas pelos pesquisadores existem certamente milhares de algoritmos. A possibilidade de alterações em métodos e parâmetros é tão extensa que há necessidade de cautela para apresentar uma *novidade*. É verdade que muito já se publicou sobre os MOEAs, mas como várias linhas de pesquisa, o assunto está longe da maturidade e, uma idéia que pode parecer óbvia, talvez ainda não tenha sido apresentada.

O NSGA (Non-Sorting Genetic Algorithm) inicialmente proposto por Srinivas e Deb em [115] tem duas finalidades principais: a) gerar as soluções não dominadas; e b) manter diversidade destas soluções. A cada geração os  $n_{pop}$  indivíduos são agrupados em subpopulações que se distribuem em fronteiras não dominadas. Técnicas de nicho buscam desenvolver estas subpopulações espalhando soluções sobre cada fronteira não dominada. Segundo, Deb *et al.* [34], os modelos de MOEAs que utilizam mecanismos de classificação baseados em não dominância e partilha tem sido criticados principalmente por causa do custo computacional  $O(n_f * n_{pop}^3)$ , abordagem não elitista e a necessidade de se especificar um parâmetro para partilha. No mesmo trabalho, Deb *et al.* amenizam estes pontos negativos lançando o NSGA-II. Para resolver o custo relacionado a classificação da população em fronteiras de pontos eficientes, o NSGA-II associa a cada solução candidata um contador para o número de soluções que a dominam, e outra contendo índices das soluções dominadas pela solução candidata. As soluções não dominadas pertencem à 1ª fronteira e são separadas do grupo. O processo continua, e as fronteiras vão sendo separadas uma a uma. Para preservar a diversidade, o NSGA-II substituiu a função de partilha por uma abordagem de *comparação de multidão*<sup>1</sup>. Sendo que esta nova aproximação não requer qualquer definição de parâmetro feita pelo usuário para manter a diversidade entre os membros da população. O elitismo é introduzido comparando as melhores soluções da nova geração com a anterior. Os vencedores são mantidos nas primeiras fronteiras.

Coello e Sierra [21] apresentam a co-evolução como a mudança na composição genética de uma espécie como resposta para mudança genética de uma outra. O relacionamento entre duas espécies diferentes pode ser des-

---

<sup>1</sup> Termo original: crowded-comparison

critério considerando todos os possíveis tipos de interação. Partindo desta idéia, as abordagens de MOEAs envolvem troca de material genético entre indivíduos de populações diferentes. O desempenho individual depende dos indivíduos de diferentes populações. Ainda segundo Coello e Sierra, existem duas classes de algoritmos co-evolucionários: a) baseados em relacionamento de competição; b) baseados em relacionamento de cooperação. O CO-MOEA (Coevolutionary Multi-Objective Evolutionary Algorithm) utiliza tanto da competição quanto da colaboração dentro de sua estrutura. Os autores do CO-MOEA dividem a execução em quatro estágios que permanecem ativos durante parte das gerações. O CO-MOEA foi comparado com o NSGA-II, com o M $\mu$ GA (Coello e Pulido [20]) e com PAES (Knowles [72]) e seus resultados foram considerados competitivos.

Soares *et al.* [114] apresentaram o EEA (Election Evolutionary Algorithm). O EEA possui a estrutura básica semelhante aos mais tradicionais algoritmos genéticos, ou seja, os indivíduos são codificados binariamente, o cruzamento é feito com apenas um ponto de corte e, a mutação ocorre como evento de mudança de um bit, segundo dada probabilidade. O que há de diferente no EEA são os processos de seleção e de separação de fronteiras: a dominância de um indivíduo não é medida em relação a todos os outros membros da população e sim com a *maioria* da população, sendo a definição de maioria dependente dos vários tipos de métodos utilizados nos processos eleitorais políticos. O parâmetro maioria tem papel fundamental para distribuição de fronteiras e conseqüentemente para convergência do algoritmo. O EEA utiliza estratégia elitista para formação da nova população, já que mantém os elementos das primeiras fronteiras das duas últimas populações.

O algoritmo PSO (Particle Swarm Optimization) foi proposto por Kenneth e Eberhart [69] em 1995. Trata-se de um algoritmo baseado no vôo dos pássaros quando estão em bando, ou similarmente, o movimento de peixes em cardumes. O líder dita a velocidade e conduz todos os companheiros. Se ele muda o rumo, todos o acompanham. Em alguns momentos há espalhamento e subitamente voltam a se reagrupar. Ainda segundo Kenneth e Eberhart, zoólogos criaram modelos sobre esta movimentação, nos quais as distâncias inter-individuais e velocidade são características importantes. A movimentação ocorre, por exemplo, para evitar predadores, procurar comida, impressionar a fêmea para acasalar, otimizar a temperatura ambiente, etc. Analogamente, cada pássaro ou peixe poderia ser uma partícula (solução). O espalhamento teria a finalidade de investigar o espaço, o agrupamento apontaria a melhor solução dentro do grupo. Trazendo essas idéias para o contexto da otimização, o espalhamento entre soluções ajuda a mapear pontos ao longo da fronteira eficiente e, a velocidade de aproximação, trata da convergência local.

Há diversos outros MOEAs que têm sido freqüentemente citados nos periódicos de computação evolucionária e que não foram descritos neste trabalho. Uma lista não exaustiva destes métodos pode ser composta pelos algoritmos: a) ENORA (Evolutionary Algorithm of Non Dominated Sorting with Radial Slots); b) DRMOGA (Divided Range Multi-Objective Genetic Algorithm); c) M $\mu$ GA (Multi-objective MicroGA) [20]; d) PAES (Pareto Archived Evolution Strategy) [72]; e e)SPEA (Strength Pareto Evolutionary Algorithm) [131].

#### 2.2.4 Parâmetros Gerais

Toda vez que um novo algoritmo é apresentado, seu desempenho é testado frente a outros já consagrados na literatura. Bons exemplos para estas comparações podem ser encontrados em Deb *et al.* [34], Knowles [72] e Zitzler *et al.* [132]. Nestes trabalhos, os autores propuseram algoritmos com estruturas diferentes, sendo que as particularidades são assistidas por uma série de parâmetros. Dependendo da variação introduzida em um destes parâmetros o desempenho geral do algoritmo fica afetado. Até mesmo a escolha do número de execuções pode gerar uma estatística de convergência diferente, visto que os MOEAs são estocásticos. Por esses e outros motivos, torna-se importante analisar quais aspectos interferem na comparação de algoritmos.

MOEAs baseados em populações e codificação binária têm como ponto de partida discretizar o espaço de busca. Se  $n_{bits}$  é o número de bits para cada faixa de variável, então o número de cadeias de caracteres possíveis é  $2^{n_{bits}}$  para cada variável. Uma forma de comparar erroneamente é discretizar o espaço com  $n_{bits}$  diferentes, como ocorreu quando Zitzler *et al.* [132] registrou resultados divergentes de Veldhuizen [121].

O tamanho da população  $n_{pop}$  interfere em praticamente todos os aspectos do MOEA. Deixado outros parâmetros fixos, população grande tem mais diversidade, é mais lenta para convergir, a realização de processos como seleção e separação de fronteiras é mais cara, despende-se mais memória para armazenamento e aumentam as chances de mapeamento de mais pontos na fronteira eficiente. Logo, ao comparar-se algoritmos com populações de tamanhos muito diferentes pode-se esconder a real situação entre um método e outro. Uma maneira de amenizar este erro é contabilizar o número de cálculos de função em cada caso.

A probabilidade de operadores de recombinação  $p_c$  e de diversidade  $p_m$  altera consideravelmente os resultados finais de execução. Fato verificado em Soares [113]. Muitas vezes, novos algoritmos trazem consigo diferentes operadores. Uma prática comum é aplicar os operadores como proposto pelos autores de cada algoritmo e ajustar a probabilidade de seus novos opera-

dores segundo o melhor desempenho adquirido em testes empíricos. Testes como os realizados em 1986 por Grefenstette [47] que encontrou  $p_c = 0.60$  e  $p_m = 0.001$  para os Algoritmos Genéticos descritos em seu artigo.

O processo de seleção de indivíduos para a etapa de recombinação afeta, por exemplo, a convergência dos MOEAs. Assim, a comparação entre algoritmos que empregam métodos de seleção diferentes, pode ser, em alguns casos, incoerente e perder sua validade.

O número  $n_f$  de funções objetivo e o número  $n_x$  de variáveis têm sido fatores determinantes de desempenho. O motivo é que em problemas multi-objetivos o cálculo de dominância e as técnicas de nicho exigem comparação aos pares de indivíduos. Logo, para que a comparação seja justa entre algoritmos, os valores de parâmetros como  $n_f$  e  $n_x$  devem ser próximos. Para manter esses parâmetros constantes basta adotar as mesmas funções teste durante as comparações.

Comparar tempo computacional não é tarefa fácil. Nos trabalhos científicos, geralmente são detalhadas as explicações sobre os algoritmos, mas nem sempre sobre as implementações realizadas. Talvez por não ficar elegante ou por ocupar espaço. No entanto, durante a implementação, soluções numéricas criativas dão resultados significativamente menores em termos de custo computacional. Rump [101] multiplicou duas matrizes **A** e **B** de dimensões  $200 \times 200$  de quatro formas diferentes, no MATLAB, utilizando um Pentium I 300Mhz. A diferença entre a implementação melhor e a pior foi 2265 vezes! Portanto, para comparar tempo computacional é necessário cautela.

Os MOEAs são sensíveis a mudança dos valores de seus parâmetros. Ao realizar um confronto entre algoritmos, deve-se observar cuidadosamente o conjunto de parâmetros de cada método para fazer o confronto mais justo.

### 2.2.5 Elitismo

Quando Zitzler [131] propôs SPEA ele o comparou com outros sete algoritmos, entre eles o NSGA, o NPGA (Niche Pareto Genetic Algorithm) e o MOGA. O desempenho do SPEA foi significativamente melhor porque era o único que tinha em sua estrutura uma estratégia elitista. Na mesma pesquisa, foi introduzido elitismo nos demais algoritmos. Todos melhoram o desempenho. Este fato não traz nenhuma surpresa para algoritmos evolucionários multi-objetivos. Uma vez encontrado um ponto eficiente, ele não pode ser descartado e, por outro lado, não se pode permitir que ele funcione como um atrator para toda população. Difícil afirmar qual estratégia é melhor. A seguir algumas implementações de elitismo.

Rudolph [99] descreveu um MOEA elitista que mantém várias soluções não dominadas para a próxima geração. Quando um valor limite de soluções

eficientes, fixado anteriormente, é atingido, não há espaço para novos pontos.

O elitismo no NSGA II é mantido simplesmente realizando comparações de dominância entre a população anterior e a atual. Os  $n_{pop}$  primeiros elementos não dominados formam a próxima população. Novos pontos podem ser adicionados em toda geração, mas há um acréscimo no custo computacional com a comparação de duas populações.

Por outro lado, o SPEA realiza elitismo simplesmente mantendo uma população externa. Desde as primeiras gerações os pontos não dominados são armazenados numa população de tamanho fixo. A cada iteração as novas soluções eficientes são classificadas junto às da população externa. A diversidade é controlada por meio do cálculo da distância via *hipercubos*. Soluções muito próximas são eliminadas da população externa.

Inserir elitismo foi um passo evolutivo e definitivo dentro dos MOEAs. Não há como afirmar que existam trabalhos que divergem desta afirmação. Mas nas bibliografias mais citadas é indiscutível que as melhores soluções devam ser guardadas de alguma forma.

### 2.2.6 Métricas para Medir Desempenho

Em Zitzler *et al.* [132], os algoritmos evolucionários multi-objetivos têm, além dos objetivos propostos no problema, mais outros três objetivos: a) encontrar o seu conjunto de soluções não dominadas o mais próximo possível da Fronteira de Pareto; b) possuir soluções uniformemente espaçadas ao longo da fronteira; e c) maximizar pontos nos extremos da fronteira não dominada para cada objetivo. De forma semelhante, Deb [32] enfatiza como meta para os MOEAs apenas os itens (a) e (b). Independente do item a ser empregado, quando deseja-se comparar MOEAs ou medir sua eficiência em resolver os problemas de otimização, uma prática freqüente em muitos estudos é apresentar os resultados em gráficos bi ou tridimensionais, onde observa-se com facilidade o comportamento e talvez até mesmo o desempenho dos algoritmos em questão. Não há dúvidas que se o resultado visual for de alguma forma, quantificado e confirmado, há suporte científico para distinguir qual a melhor opção. No entanto, se o problema de otimização possuir vários objetivos a apresentação dos resultados em forma gráfica torna-se mais complicada.

Com a intenção de investigar desempenho de algoritmos, pesquisadores como Czyzak [24], Deb [32], Hansen e Jaszkiewicz [51], Schott [104], Veldhuizen [121], Zitzler [131] e Zitzler *et al.* [132] apresentaram trabalhos relevantes para construção de *métricas* para MOEAs. Algumas delas estão descritas nesta seção. Mas, antes de apresentá-las, alguns conceitos importantes devem ser introduzidos.

Considere inicialmente os conjuntos de soluções não dominadas  $\mathbf{A} \subset$

$\mathbb{R}^{n_n}$  e  $\mathbf{B} \subset \mathbb{R}^{n_n}$  de dois MOEAs quaisquer, sendo  $n_n$  uma dimensão genérica. Agora, considere o operador  $\Upsilon_{\prec}$  represente uma função que retorne o subconjunto de pontos não dominados de um conjunto “.” (a definição formal de  $\Upsilon_{\prec}$  é apresentada na Subseção 3.2.1). De acordo com Hansen e Jaszekiewicz [51], o conjunto solução  $\mathbf{A}$  pode ser declarado melhor que  $\mathbf{B}$  em três níveis: a)  $\mathbf{A} \prec_f \mathbf{B}$  ( $\mathbf{A}$  é fracamente melhor que  $\mathbf{B}$ ); b)  $\mathbf{A} \prec_F \mathbf{B}$  ( $\mathbf{A}$  é fortemente melhor que  $\mathbf{B}$ ); e c)  $\mathbf{A} \prec_C \mathbf{B}$  ( $\mathbf{A}$  é completamente melhor que  $\mathbf{B}$ ). Abaixo, encontram-se as expressões que definem essas conclusões sobre  $\mathbf{A}$  e  $\mathbf{B}$ , e na Fig. 2.2 encontram-se os exemplos gráficos.

$$\mathbf{A} \prec_f \mathbf{B} \Leftrightarrow \Upsilon_{\prec}(\mathbf{A} \cup \mathbf{B}) = \mathbf{A}, \quad \mathbf{A} \neq \mathbf{B} \quad (2.9)$$

$$\mathbf{A} \prec_F \mathbf{B} \Leftrightarrow \Upsilon_{\prec}(\mathbf{A} \cup \mathbf{B}) = \mathbf{A}, \quad \mathbf{B} \setminus \Upsilon_{\prec}(\mathbf{A} \cup \mathbf{B}) \neq \emptyset \quad (2.10)$$

$$\mathbf{A} \prec_C \mathbf{B} \Leftrightarrow \Upsilon_{\prec}(\mathbf{A} \cup \mathbf{B}) = \mathbf{A}, \quad \mathbf{B} \cap \Upsilon_{\prec}(\mathbf{A} \cup \mathbf{B}) = \emptyset \quad (2.11)$$

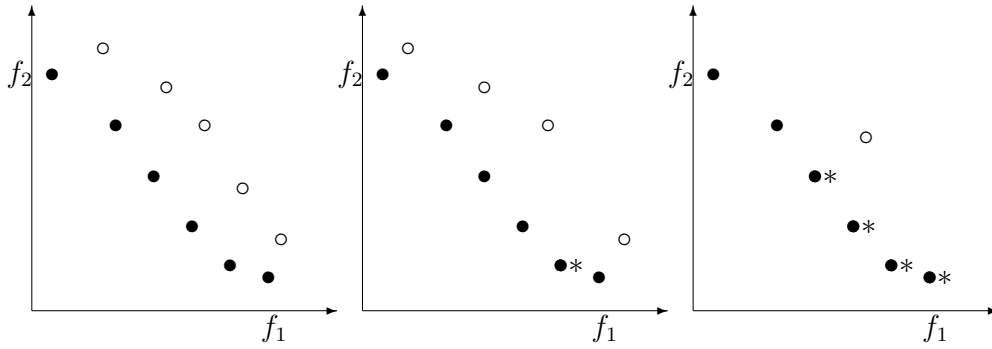


Fig. 2.2: Relação de desempenho entre os conjuntos de solução  $\mathbf{A}$ ( $\bullet$ ) e  $\mathbf{B}$ ( $\circ$ ). No primeiro caso,  $\mathbf{A} \prec_C \mathbf{B}$ . Na sequência,  $\mathbf{A} \prec_F \mathbf{B}$  e  $\mathbf{A} \prec_f \mathbf{B}$ . O símbolo \* representa ponto comum em  $\mathbf{A}$  e  $\mathbf{B}$ . Nota-se claramente que se  $\mathbf{A} \prec_C \mathbf{B} \Rightarrow \mathbf{A} \prec_F \mathbf{B} \wedge \mathbf{A} \prec_f \mathbf{B}$ .

As métricas para comparar os resultados de algoritmos multi-objetivos podem não estar em sintonia com os conceitos apresentados em (2.9), (2.10) e (2.11). Por isso dois critérios de compatibilidade foram propostos no mesmo estudo.

**Definição 2** *Compatível fracamente:* uma métrica de comparação é fracamente compatível com a relação de desempenho  $\prec_i$ ,  $i \in \{f, F, C\}$  se para cada par de conjuntos não dominados  $\mathbf{A}$  e  $\mathbf{B}$ , tal que  $\mathbf{A} \prec_i \mathbf{B}$ , a métrica avalia  $\mathbf{A}$  como não pior que  $\mathbf{B}$ .

**Definição 3** *Compatível*: uma métrica de comparação é compatível com a relação de desempenho  $\prec_i$ ,  $i \in \{f, F, C\}$  se para cada par de conjuntos não dominados  $\mathbf{A}$  e  $\mathbf{B}$ , tal que  $\mathbf{A} \prec_i \mathbf{B}$ , a métrica avalia  $\mathbf{A}$  melhor que  $\mathbf{B}$ .

As métricas apresentadas a seguir são avaliadas quanto à compatibilidade com  $\prec_C$ ,  $\prec_F$  e  $\prec_f$ , quanto a utilizarem mecanismos para classificar as soluções, quanto à necessidade de comparação com conjuntos Pareto e quanto ao custo computacional. O estudo destas métricas está suportado principalmente pelo livro de Deb [32] e pela tese de doutorado de Knowles [72]. Informações complementares podem ser obtidas nas referências originais.

**Taxa de Erro**: Com a finalidade de contabilizar a proporção de pontos eficientes, Veldhuizen [121] elaborou a métrica TE (Taxa de Erro)<sup>2</sup> que é baseada no conhecimento prévio dos pontos eficientes  $\mathbf{Y}^*$  como mostrado na equação abaixo.

$$TE = \frac{\sum_{i=1}^{|\mathbf{A}|} e_i}{|\mathbf{A}|}, \quad (2.12)$$

sendo  $|\mathbf{A}|$  o número de soluções de  $\mathbf{A}$ , e contador de erros  $e_i$  é tal que:

$$\begin{cases} e_i = 1, & \text{se } e_i \in \mathbf{A} \setminus \mathbf{Y}^*; \\ e_i = 0, & \text{caso contrário.} \end{cases} \quad (2.13)$$

Para concluir se um ponto pertence ou não a  $\mathbf{Y}^*$ , definiu-se uma distância Euclideana mínima com relação a qualquer um dos elementos pertencentes ao conjunto referência  $\varepsilon_{TE}$ . Em outras palavras,  $\varepsilon_{TE}$  representa a precisão da métrica TE medida em termos de distância Euclideana.

TE é uma métrica simples de implementar, com poucos parâmetros e é simétrica, ou seja,  $(1 - ER)$  contabiliza o número de pontos pertencentes a  $\mathbf{Y}^*$ . Como desvantagens, têm-se: a) o custo computacional associado à comparação de proximidade com ponto eficiente em  $\mathbf{Y}^*$ ; b) não mede uniformidade espacial das soluções em  $\mathbf{A}$ ; c) penaliza conjuntos soluções com muitos pontos em  $\mathbf{Y}^*$ , se também existirem muitos pontos fora; e d) a métrica é fracamente compatível com  $\prec_C$ .

Importante observar que a métrica TE foi desenvolvida para ser aplicada no espaço dos objetivos. Analogamente, ela pode medir a proximidade do conjuntos solução encontrado em relação ao conjunto solução referência, no espaço de busca.

---

<sup>2</sup> Termo original: Error Ratio

**Métrica  $\mathcal{C}$ :** A métrica  $\mathcal{C}$  foi proposta por Zitzler [131] para comparar dois conjuntos de soluções não dominadas  $\mathbf{A}$  e  $\mathbf{B}$  quanto à dominância. Formalmente,

$$\mathcal{C}_{\mathbf{A},\mathbf{B}} = \frac{|\{b \in \mathbf{B} \mid \exists a \in \mathbf{A} : a \preceq b\}|}{|\mathbf{B}|} \quad (2.14)$$

Considerando-se dois conjuntos  $\mathbf{A}$  e  $\mathbf{B}$  quaisquer, observa-se que não necessariamente  $\mathcal{C}_{\mathbf{A},\mathbf{B}} = \mathcal{C}_{\mathbf{B},\mathbf{A}}$ .

O custo computacional é baixo quando comparado às demais métricas desta seção, mesmo sendo preciso calcular  $\mathcal{C}_{\mathbf{A},\mathbf{B}}$  e  $\mathcal{C}_{\mathbf{B},\mathbf{A}}$ . Também é uma métrica simples de implementar e de interpretar, não exigindo a classificação do conjunto solução e não necessitando da informação do conjunto  $\mathbf{Y}^*$ . No entanto, essa métrica é compatível apenas com  $\prec_C$  e ainda ela não mede uniformidade da distribuição das soluções.

**Métrica do Espaçamento:** Diferentemente da finalidade das métricas anteriores, Schott [104] elaborou um mecanismo para medir a qualidade da distribuição dos pontos ao longo da fronteira não dominada. Considerando novamente que  $|\mathbf{A}|$  representa o número de soluções de  $\mathbf{A}$ , observe abaixo como o cálculo de uniformidade é feito.

$$\mathcal{S} = \sqrt{\frac{1}{|\mathbf{A}|} \sum_{i=1}^{|\mathbf{A}|} (d_i - \bar{d})^2} \quad (2.15)$$

O termo  $d_i$  é a menor distância, no espaço dos objetivos, a qualquer outro ponto pertencente a  $\mathbf{A}$  e  $\bar{d}$  é a média de todos os  $d_i$  calculados. Esta formulação simplesmente calcula o desvio padrão das menores distâncias entre soluções. Logo, quanto menor o desvio padrão mais uniformemente estão espalhados os pontos.

Como vantagens, a métrica de Schott não depende de comparação com conjunto Pareto e não necessita classificar o conjunto  $\mathbf{A}$ . E como desvantagens, a métrica não mede a proximidade com a fronteira eficiente, tem custo computacional relativamente alto e ainda não é possível definir relação de compatibilidade com  $\prec_f$ ,  $\prec_F$  ou  $\prec_C$ .

**Métrica do Espalhamento :** Trata-se de outra métrica destinada a medir qualidade na distribuição das soluções que foi desenvolvida por Deb *et al.* e descrita em [32]. Seu cálculo é mostrado a seguir.

$$\Delta = \frac{\sum_{m=1}^M d_m^e + \sum_{i=1}^{|\mathbf{A}|} |d_i - \bar{d}|}{\sum_{m=1}^M d_m^e + |\mathbf{A}| \bar{d}} \quad (2.16)$$

Sendo  $\Delta$  o valor da métrica,  $d_i$  é a distância até algum outro ponto na vizinhança,  $\bar{d}$  é a média de todos os  $d_i$  calculados e  $d_m^e$  é a distância entre os pontos extremos de  $\mathbf{A}$  e  $\mathbf{Y}^*$  em cada objetivo. Portanto, são investigadas tanto a uniformidade de distribuição dos pontos quanto o quão extensa é essa distribuição na fronteira. Na implementação desta métrica, realizada neste trabalho, calculou-se  $d_i$  como a menor distância entre uma solução  $i$  a qualquer outra do conjunto solução.

Uma vantagem desta métrica é a penalização de conjuntos não dominados com soluções dispersas e longe dos extremos em cada objetivo. Por outro lado, são consideradas como desvantagens: a) a necessidade de se conhecer  $\mathbf{Y}^*$ ; b) o custo com classificação tanto de  $\mathbf{A}$  quanto de  $\mathbf{Y}^*$  para todos os objetivos; c) a incompatibilidade com as relações de desempenho; e d) a não medição de proximidade com a fronteira de Pareto.

**Métrica do Hipervolume:** Essa métrica é endereçada tanto para analisar proximidade de conjuntos solução à fronteira de Pareto quanto para qualificar a uniformidade da distribuição dos pontos sobre a fronteira. Cada ponto do conjunto solução  $\mathbf{A}$  é um vértice de um hipercubo  $hc_i$ . O outro vértice, tomando-se a diagonal principal, é um ponto fixo que pode ser definido como o pior valor para cada objetivo. Considerando um conjunto solução  $\mathbf{A}$ , a união das regiões de cada hipercubo gera um hipervolume HV que é normalizado com um hipervolume HVR construído a partir de  $\mathbf{Y}^*$ . Observa-se que o índice é acrescentado para denotar o conjunto sobre o qual age a métrica do hipervolume. As equações (2.17) e (2.18) deixam claro como o cálculo do hipervolume é feito.

$$HV_A = volume\left(\bigcup_{i=1}^{|\mathbf{A}|} hc_i\right) \quad (2.17)$$

$$HVR_A = \frac{HV_{\mathbf{A}}}{HV_{\mathbf{Y}^*}} \quad (2.18)$$

Pelas expressões acima, conclui-se que hipervolume maior significa melhor proximidade da fronteira eficiente.

A métrica do hipervolume tem custo computacional relativamente médio quando comparado às outras métricas, é de fácil implementação e interpretação para casos 2D e quantifica tanto dispersão quanto proximidade de  $\mathbf{Y}^*$ . Entretanto, a classificação das soluções e a escolha do ponto fixo pode não ser tarefa trivial. Adicionalmente, a compatibilidade da métrica do hipervolume com  $\prec_f$ ,  $\prec_F$  ou  $\prec_C$  não está clara.

Como descrito, todas as métricas apresentam vantagens e desvantagens para medir a proximidade com a fronteira de Pareto e uniformidade das

soluções. Certamente, o uso de mais de uma métrica ajudar a julgar melhor os conjuntos soluções encontrados. Para os interessados em aprofundar no estudo de métricas, sugere-se a consulta Knowles [72].

### 2.2.7 Técnicas de Nicho

Diversidade com qualidade é umas das características chaves buscadas pelos MOEAs. Há algum tempo, época dos primeiros MOEAs, como o SGA para problema mono-objetivo, a convergência prematura para uma determinada solução já não era desejada. O motivo principal é que a perda da diversidade dificulta a exploração de novas regiões e, conseqüentemente o algoritmo pode ficar “preso” em uma região ótima local. Técnicas de manutenção de diversidade, conhecidas como *técnicas de nicho*, foram sendo desenvolvidas para tentar resolver problemas desta natureza. A seguir, são descritos três modelos destes procedimentos.

A primeira técnica de nicho para os MOEAs foi desenvolvida por Goldberg e Richardson [46] em 1987. A idéia era alterar, o valor da função objetivo de acordo com o número de indivíduos que estivessem na vizinhança. Uma *função de partilha*, como por exemplo (2.19), mede a proximidade entre duas soluções.

$$s_{d_{ij}} = \begin{cases} (1 - \frac{d}{\sigma_{partilha}})^\alpha, & \text{se } d_{ij} \leq \sigma_{partilha} \\ 0, & \text{caso contrário.} \end{cases} \quad (2.19)$$

Sendo  $d_{ij}$  a distância entre as soluções  $i$  e  $j$ . O somatório de todas as funções de partilha, define o quanto o desempenho  $f_i$  do indivíduo  $i$  será degradado. O novo desempenho  $f'_i$  segundo a técnica de nicho é dado por  $f'_i = f_i / \sum s_{d_{ij}}$ .

A função de partilha de Goldberg e Richardson foi adaptada para problemas multi-objetivos em alguns algoritmos, como é o caso do WBGA (Weight-Based Genetic Algorithm). Nesta abordagem o valor dos pesos são definidos via (2.19). Algoritmos multi-objetivos baseados em pesos tem problemas quando a fronteira de Pareto é não convexa. Mas em casos convexos WBGA consegue distribuir pontos ao longo da fronteira.

O NSGA II substitui a função de partilha por um modelo onde o parâmetro  $\sigma_{partilha}$  não necessita ser determinado. Em primeiro lugar uma *estimativa de densidade populacional* é calculada, através do cálculo de um cubóide formado com vértices nas soluções mais próximas. O perímetro do cubóide é denominado de *distância de agrupamento*. Em seguida um operador de comparação de agrupamento guia o processo de seleção levando em conta uma classificação de não dominância e na distância calculada no procedimento de estimativa de densidade.

Além das técnicas de nicho, um algoritmo que cria *aglomerados*<sup>3</sup> foi proposto por Zitzler e Thiele [133] com a mesma finalidade de dificultar a concentração de soluções muito próximas. Segundo os autores, os resultados apresentados também foram satisfatórios.

### 2.3 Análise Intervalar

Hayes [53] e Daumas [29] relataram alguns acidentes ocorridos com equipamentos de tecnologia avançada que resultaram em vítimas fatais. Foram registrados como causa, os erros em rotinas de cálculo numérico, devido por exemplo, à propagação desastrosa de arredondamentos. Qualquer um, ciente apenas dos fatos, poderia dizer que as rotinas numéricas aplicadas pudessem ter sido construídas descuidadamente ou que os hardware utilizados deveriam ter sido mais precisos e confiáveis. Sabendo-se que hardware trabalha com grandezas discretas, conclui-se que ele tem limites finitos para tratar valores precisos. O número de acidentes desta natureza pode aumentar com a crescente aplicação de tecnologia de precisão, habitualmente empregada em conjunto com a automação de processos. Em contra-partida, novos mecanismos têm sido criados para aumentar a segurança, oferecer um nível mais alto de confiabilidade diminuindo os riscos de danos materiais e humanos. Algumas destas soluções são baseadas em métodos intervalares, definidos em novo conjunto numérico próprio, o conjunto dos Números Intervalares Reais  $\mathbb{IR}$ . Este conjunto manipula incertezas considerando as grandezas numéricas como entidades compreendidas dentro de uma faixa de valores.

A Análise Intervalar, área que estuda o conjunto dos Números Intervalares, começou com Moore [85] justamente para tratamento de erros de arredondamento em computação numérica. Atualmente, ela tem sido eficientemente utilizada em aplicações que envolvem parâmetros de incerteza, veja Jaulin et al. [63] e Kearfott e Kreinovich [68]. Uma manipulação segura destes intervalos resulta em um intervalo no qual se pode garantir a presença da solução pontual. Entretanto, os métodos intervalares sofrem algumas críticas quanto à lentidão, em termos de tempo computacional. Hansen e Walster [50] rebateram a maioria das críticas e destacam as vantagens dos métodos intervalares: a) a possibilidade de obter soluções de certos problemas que não podem ser resolvidos por métodos não intervalares, como por exemplo, os problemas que envolvem operações de divisão por zero em sua formulação; b) a convergência, pois as etapas dos processos internos dos métodos intervalares são finitas; e c) a confiabilidade dos resultados. Uma outra característica importante a ser inserida no conjunto de vantagens: a

---

<sup>3</sup> Termo original: clustering

usual globalidade dos métodos intervalares.

Análise Intervalar tem sido empregada em diversas áreas como por exemplo: Carreras e Walker [15] e Morales e Son [88] em robótica; Soares *et al.* [111] e Siouris *et al.* [106] em rastreamento por radar;

### 2.3.1 Breve Histórico

É difícil precisar onde os primeiros indícios sobre intervalos surgiram. Alguns estudos apontam que provavelmente o primeiro documento onde aparece a presença de intervalos foi elaborado por Arquimedes, registrado no livro de Healt [54] em 1897, onde Arquimedes buscava calcular o valor de  $\pi$ . No entanto, existe divergência entre os pesquisadores da área de Análise Intervalar quando dissertam sobre quem efetuou os passos concretos iniciais. Alefeld e Mayer [2] registraram alguns estudos isolados na área de Computação Científica, na década de 1950. Sunaga [117] em 1958, apresentou as primeiras regras sobre operações aritméticas intervalares simples e também para vetores e matrizes. Sunaga ainda tratou da resolução de equações intervalares e cálculo de funções integrais. Em 1965 Hansen [48], realizou manipulação de intervalos com a álgebra linear e, no mesmo ano, um grupo de pesquisadores alemães incluindo Alefeld, Krawczyk e Nickel desenvolveram vários aspectos de implementações computacionais. Embora todos os estudos acima sejam relevantes, o primeiro trabalho consistente, reconhecido por todos os pesquisadores como uma contribuição real para a Análise Intervalar foi o livro de Moore [85] em 1966. Durante quase duas décadas, a Análise Intervalar não foi um tema muito investigado pela comunidade científica. Talvez, simplesmente porque ainda não existia necessidade para sua utilização. No fim deste período, trabalhos interessantes foram realizados por Neumaier [90], em 1985, sobre o conjunto solução de equações lineares e não lineares e, Kearfott [67] sobre otimização, em 1989. Nos anos 90, a Análise Intervalar tornou-se popular com a adesão de vários grupos de pesquisa nas mais diversas partes do mundo.

Retornando a 1966, o mérito maior de Moore [85] foi devido à organização realizada na pesquisa sobre intervalos até aquele momento, fundamentando e transformando a Aritmética em Análise Intervalar. No entanto, situações que envolviam intervalos contendo 0 e limitados por  $\pm\infty$  não tinham solução. Para resolver este problema, Hanson [52] e Kahan [66], ambos em 1968, em pesquisas independentes, descreveram incompletas extensões nas quais os limites para os intervalos pudessem ter valor  $\pm\infty$ . Estes estudos marcaram o início do Sistema Intervalar Fechado, que é uma extensão do Sistema Intervalar Finito proposto por Moore [85]. Mais tarde, Walster *et al.* em [123] descreveram um sistema que representasse completamente uma extensão do

sistema de Moore. Neste modelo, a resolução de problemas que envolvam as operações  $\frac{0}{0}$ ,  $\infty - \infty$ ,  $0 \times \infty$  e  $\frac{\pm\infty}{\pm\infty}$  foram definidas.

### 2.3.2 Relações Algébricas de Conjuntos

Números Intervalares são definidos como conjuntos, e portanto, ele herda as operações e relações para esse tipo de dado. A seguir, uma revisão baseada em Jaulin *et al.* [63] sobre relações básicas entre conjuntos é apresentada, e na seqüência, as operações baseadas nessas relações.

Considere dois conjuntos numéricos  $\mathbf{A}$  e  $\mathbf{B}$  quaisquer. São válidas as seguintes operações, relações e definições:

$$\text{Intersecção} : \mathbf{A} \cap \mathbf{B} = \{a \mid a \in \mathbf{A} \wedge a \in \mathbf{B}\}. \quad (2.20)$$

$$\text{União} : \mathbf{A} \cup \mathbf{B} = \{a \mid a \in \mathbf{A} \vee a \in \mathbf{B}\}. \quad (2.21)$$

$$\text{Complemento} : \mathbf{A} \setminus \mathbf{B} = \{a \mid a \in \mathbf{A} \wedge a \notin \mathbf{B}\}. \quad (2.22)$$

$$\text{Produto Cartesiano} : \mathbf{A} \times \mathbf{B} = \{(a, b) \mid a \in \mathbf{A} \wedge b \in \mathbf{B}\}. \quad (2.23)$$

$$\text{Inclusão} : \mathbf{A} \subset \mathbf{B} \Leftrightarrow \forall a \in \mathbf{A}, a \in \mathbf{B}. \quad (2.24)$$

$$\text{Igualdade} : \mathbf{A} = \mathbf{B} \Leftrightarrow \mathbf{A} \subseteq \mathbf{B} \wedge \mathbf{B} \subseteq \mathbf{A}. \quad (2.25)$$

Se  $\mathbf{Z} = \mathbf{A} \times \mathbf{B}$  e  $\mathbf{Z}_1 \subset \mathbf{Z}$ , então a projeção de  $\mathbf{Z}_1$  sobre  $\mathbf{A}$ , com respeito a  $\mathbf{B}$ , é dada por:

$$\text{proj}_{\mathbf{A}}(\mathbf{Z}_1) = \{a \in \mathbf{A}, \exists b \in \mathbf{B} \mid (a, b) \in \mathbf{Z}_1\}. \quad (2.26)$$

A Fig.2.3 ilustra o produto cartesiano e a projeção de conjuntos.

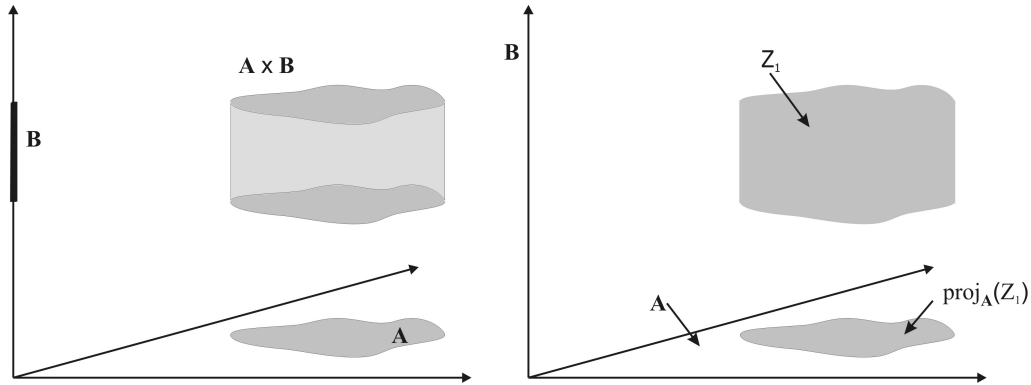


Fig. 2.3: Operações sobre conjuntos: produto cartesiano e projeção.

Novamente, considere dois conjuntos  $\mathbf{A}$  e  $\mathbf{B}$  e uma função  $f : \mathbf{A} \rightarrow \mathbf{B}$ . Se  $\mathbf{A}_1 \subset \mathbf{A}$ , a *imagem direta* de  $\mathbf{A}_1$  por  $f$  é:

$$f(\mathbf{A}_1) = \{f(a) \mid a \in \mathbf{A}_1\}. \quad (2.27)$$

Se  $\mathbf{B}_1 \in \mathbf{B}$ , a *imagem recíproca* de  $\mathbf{B}_1$  para  $f$  é

$$f^{-1}(\mathbf{B}_1) = \{a \in \mathbf{A} \mid f(a) \in \mathbf{B}_1\}. \quad (2.28)$$

Se  $\emptyset$  representa o conjunto vazio, então:

$$f(\emptyset) = f^{-1}(\emptyset) = \emptyset. \quad (2.29)$$

Partindo das expressões (2.27), (2.28) e (2.29) e definindo  $\mathbf{A}_1$  e  $\mathbf{A}_2$  como subconjuntos de  $\mathbf{A}$ , infere-se que:

$$\begin{aligned} f(\mathbf{A}_1 \cap \mathbf{A}_2) &\subseteq f(\mathbf{A}_1) \cap f(\mathbf{A}_2). \\ f(\mathbf{A}_1 \cup \mathbf{A}_2) &= f(\mathbf{A}_1) \cup f(\mathbf{A}_2). \\ f^{-1}(\mathbf{A}_1 \cap \mathbf{A}_2) &= f^{-1}(\mathbf{A}_1) \cap f^{-1}(\mathbf{A}_2). \\ f^{-1}(\mathbf{A}_1 \cup \mathbf{A}_2) &= f^{-1}(\mathbf{A}_1) \cup f^{-1}(\mathbf{A}_2). \\ f(f^{-1}(\mathbf{A})) &\subseteq \mathbf{A}. \\ f^{-1}(f(\mathbf{A})) &\supseteq \mathbf{A}. \\ \mathbf{A}_1 \subseteq \mathbf{A}_2 &\Rightarrow f(\mathbf{A}_1) \subseteq f(\mathbf{A}_2). \\ \mathbf{A}_1 \subseteq \mathbf{A}_2 &\Rightarrow f^{-1}(\mathbf{A}_1) \subseteq f^{-1}(\mathbf{A}_2). \end{aligned} \quad (2.30)$$

### 2.3.3 Fundamentos e Definições

Algumas definições sobre conjuntos apresentadas por Hickey *et al.* [56] para suportar a teoria de intervalos são apresentadas a seguir.

**Definição 4** *Um conjunto aberto básico de reais é da forma  $\{x \in \mathbb{R} \mid a < x < b\}$ , sendo  $a \in \mathbb{R}$  e  $b \in \mathbb{R}$ . Um conjunto  $\mathbf{A}$  é aberto se para cada ponto  $x \in \mathbf{A}$  existe um conjunto aberto básico  $\mathbf{A}_1$  tal que  $x \in \mathbf{A}_1 \subset \mathbf{A}$ .*

**Definição 5** *O conjunto  $\mathbf{A}$  é fechado se seu complemento é aberto. Por definição,  $\emptyset$  conjunto é fechado.*

**Definição 6** *Um conjunto de número reais  $\mathbf{A}$  é dito conectado se não existem conjuntos separados, abertos não vazios  $\mathbf{A}_1$  e  $\mathbf{A}_2$  que intersectam  $\mathbf{A}$  e para os quais  $\mathbf{A} \subset \mathbf{A}_1 \cup \mathbf{A}_2$ . Por definição, o conjunto  $\emptyset$  é conectado.*

**Definição 7** *Um intervalo conectado fechado  $[x]$  é definido por*

$$[\underline{x}, \bar{x}] = \{x \in \mathbb{R} \mid \underline{x} \leq x \leq \bar{x}\}, \quad \underline{x} \in \mathbb{R} \text{ e } \bar{x} \in \mathbb{R}, \quad (2.31)$$

*sendo  $\underline{x}$  e  $\bar{x}$  os limites inferior e superior do intervalo  $[x]$  respectivamente. O conjunto de todos os intervalos do tipo (2.31) é  $\mathbb{IR}$ .*

Em se tratando de intervalos, alguns casos especiais são:

- intervalos tais que  $\bar{x} = \underline{x}$  são ditos *degenerados* ou *pontuais*;
- intervalo que não contém número é denominado de intervalo vazio;
- notações de intervalo como  $[0, \infty]$  e  $[0, \infty[$  tem o mesmo significado;
- as formas  $[\infty, \infty]$  e  $[-\infty, -\infty]$  não correspondem a intervalos.

Estes casos são exceções que, se não forem previstos e tratados adequadamente, deixam inválidos os teoremas, as relações e as operações intervalares que estão apresentadas adiante.

Em [120], Vaccaro descreve duas semânticas para interpretar e tratar intervalos. Na primeira, os intervalos são descritos conforme a Def. 7 e na segunda, como *Número Intervalo*. A definição formal de número intervalo é suprimida para evitar confusão e adição de nomenclatura desnecessária. Mas informalmente, o número intervalo é um tipo de dado constituído por todos os números reais pertencentes no intervalo. Assim, o intervalo  $[-1, 3]$  representa todos os números reais compreendidos entre  $-1$  e  $3$ , inclusive  $-1$  e  $3$ . Como consequência, um número intervalo difere dos reais por poder assumir simultaneamente valores negativos, positivos e nulos. Os números intervalos não são objeto de estudo deste trabalho e por isso o termo intervalo estará sempre vinculado ao tipo de dado da Def. 7.

### 2.3.4 Operações Aritméticas Elementares

Uma vez diferenciadas as semânticas sobre intervalo, o próximo passo é a definição das operações intervalares.

**Definição 8** *Considere  $\diamond$  um operador binário que represente as quatro operações clássicas da aritmética real  $+$ ,  $-$ ,  $*$  e  $/$ . Então*

$$[x] \diamond [y] = \{x \diamond y \mid x \in [x], y \in [y]\} \quad (2.32)$$

Por simplicidade, optou-se por descrever as operações aritméticas (2.32) para  $\mathbb{IR}$ . Similarmene, algumas outras definições seguem o mesmo caminho. Por analogia, o espaço  $\mathbb{IR}$  pode ser estendido a um espaço  $\mathbb{IR}^{n_x}$  qualquer.

Cada intervalo não degenerado contém infinitos números. Logo, operações aritméticas, como soma e multiplicação, devem ser diferentes daquelas utilizadas para números reais. Da mesma forma que os argumentos, a resposta também é um intervalo. Adiante, os procedimentos para cálculo das

operações básicas são descritos considerando dois intervalos  $[x]$  e  $[y]$  quaisquer. Os resultados das operações devem compreender todas as possíveis instâncias dos intervalos  $[x]$  e  $[y]$ . Assim, definem-se as operações como

$$[x] + [y] = [\underline{x} + \underline{y}, \bar{x} + \bar{y}]. \quad (2.33)$$

$$[x] - [y] = [\underline{x} - \bar{y}, \bar{x} - \underline{y}]. \quad (2.34)$$

$$[x] * [y] = [\min\{\underline{x} * \underline{y}, \underline{x} * \bar{y}, \bar{x} * \underline{y}, \bar{x} * \bar{y},$$

$$\max\{\underline{x} * \underline{y}, \underline{x} * \bar{y}, \bar{x} * \underline{y}, \bar{x} * \bar{y}\}]. \quad (2.36)$$

$$[x]/[y] = [x] * (1/[y]), \quad (2.37)$$

sendo

$$\begin{aligned} 1/[y] &= \emptyset, & \text{se } [y] = [0, 0]; \\ &= [1/\bar{y}, 1/\underline{y}], & \text{se } 0 \notin [y]; \\ &= [1/\bar{y}, \infty], & \text{se } \underline{y} = 0 \wedge \bar{y} > 0; \\ &= [-\infty, 1/\underline{y}], & \text{se } \underline{y} < 0 \wedge \bar{y} = 0; \\ &= [-\infty, \infty], & \text{se } \underline{y} < 0 \wedge \bar{y} > 0. \end{aligned} \quad (2.38)$$

Observe que a operação divisão foi construída sobre a multiplicação do numerador pelo inverso do denominador. Segundo Hickey *et al.* [56], na literatura é indiscutível a aplicação de (2.32) para os operadores  $+$ ,  $-$  e  $*$ , independente dos valores reais que os limites podem assumir. No entanto, existem divergências sobre a divisão de intervalos, quando o denominador contém zero entre seus limites inferior e superior. Definindo-se intervalos como fechados, a associação das equações (2.37) e (2.38) tem sido usualmente empregada.

Computacionalmente, algumas operações aritméticas anteriormente descritas podem ser implementadas mais eficientemente que no formato didático apresentado. Por exemplo, se antes da operação multiplicação identificar a negatividade dos limites de cada intervalo, haveria apenas dois produtos para se calcular, eliminando-se ainda as funções  $\min$  e  $\max$ . Para exemplificar as operações aritméticas descritas até então, observe os casos no Exem. 1.

### Exemplo 1 (Operações intervalares)

$$\begin{aligned} [-2, 4] + [1, 6] &= [-1, 10] \\ [-1, 5.4] - [-3, -2.1] &= [1.1, 8.4] \\ [-4, -1] * [-2, 3] &= [-12, 8] \\ [2, 3]/[2, 5] &= [0.4, 1.5] \\ [1, 3]/[0, 0] &= \emptyset \\ [1, 4]/[0, 3] &= [1/3, \infty] \\ [2, 5]/[-3, 0] &= [-\infty, -2/3] \\ [1, 2]/[-2, 1] &= [-\infty, \infty] \end{aligned}$$

## 2.3.5 Funções Intervalares Elementares

Uma *função intervalar*  $f$  é uma função cujos argumentos e resultados são intervalos. Assim, se  $f : \mathbb{IR} \rightarrow \mathbb{IR}$ , então

$$f([x]) = [\{f(x) \mid x \in [x]\}] \quad (2.39)$$

As operações aritméticas intervalares são suportadas pela (2.32). Similarmente, as funções intervalares precisam de apoio teórico para definição de seus escopos. Moore [87] descreve um teorema sobre as funções intervalares, cujas conseqüências são descritas a seguir.

Considere uma função real  $f(x_1, \dots, x_n) \rightarrow y$  e uma função intervalar  $f'([x_1], \dots, [x_n]) \rightarrow [y]$  com a mesma expressão formal da função  $f$  real. Se  $f'(x_1, \dots, x_n) = f(x_1, \dots, x_n)$  para quaisquer argumentos reais, então  $f'$  é denominada uma *função extensão intervalar* de  $f$ . Entretanto, na prática é possível que se compute apenas uma função intervalar  $f'' \supseteq f'$ , considerando as múltiplas ocorrências de seus argumentos e/ou os arredondamentos realizados nos limites dos intervalos ao longo da computação da função intervalar.

Além das operações matemáticas elementares, a expressão de  $[f]$  pode ser composta de funções matemáticas como *coseno*, *seno*, *quadrado*, *logaritmo*, etc. Na implementação computacional, os algoritmos de funções intervalares são desenvolvidos para evitar que o domínio seja violado. Veja o caso da função potência intervalar para  $n$  não negativo.

$$\begin{aligned} [x]^n &= [1, 1], & \text{se } n = 0; \\ &= [\underline{x}^n, \bar{x}^n], & \text{se } \underline{x} \geq 0 \vee n \text{ ímpar}; \\ &= [\bar{x}^n, \underline{x}^n], & \text{se } \bar{x} \leq 0 \wedge n \text{ par}; \\ &= [0, \max(\underline{x}^n, \bar{x}^n)], & \text{se } \underline{x} \leq 0 \leq \bar{x} \wedge n \text{ par}. \end{aligned} \quad (2.40)$$

O Exem. 2 trata numericamente algumas funções matemáticas.

**Exemplo 2 (Algumas funções intervalares)**

$$\begin{aligned} [-1, 3]^2 &= [0, 9] \\ \sqrt{[-10, 4]} &= [0, 2] \\ \log([-2, -1]) &= \emptyset \\ \exp^{[2, 5]} &= [7.39, 148.41] \\ \text{sen}([0, \pi]) &= [0, 1] \end{aligned}$$

Algoritmos para funções como *seno* precisam de uma atenção especial. Dependendo do intervalo, os valores da função  $f$  podem passar por vários máximos e mínimos locais que não sejam obteníveis apenas pela avaliação da

função em  $\underline{x}$  e  $\bar{x}$ . A Tab.2.4 mostra os passos básicos para construção da função  $\text{sen}([x])$ . A extensão, em termos de linhas de código, do algoritmo do

Tab. 2.4: Cálculo de  $f([x]) = \text{sen}([x])$  (Jaulin et al. [63]).

	<b>entrada</b> ( $[x]$ )
	<b>saída</b> ( $f([x])$ )
1	se $\exists k \in \mathbf{Z} \mid (2k\pi - \pi/2) \in [x]$ ,
2	então $\underline{f}([x]) = -1$ .
3	senão $\underline{f}([x]) = \min(\text{sen}(\underline{x}), \text{sen}(\bar{x}))$ .
4	se $\exists k \in \mathbf{Z} \mid (2k\pi + \pi/2) \in [x]$ ,
5	então $\bar{f}([x]) = 1$ .
6	senão $\bar{f}([x]) = \max(\text{sen}(\underline{x}), \text{sen}(\bar{x}))$ .

cálculo do seno de intervalos é um exemplo para a principal crítica sobre os métodos intervalares: o tempo computacional.

### 2.3.6 Dependência

As operações aritméticas e funções intervalares enfrentam problemas quando há repetição de uma mesma variável como em  $[x] - [x]$  e  $[x]/[x]$ , por exemplo. Para números reais, os resultados da subtração e da divisão de um número por ele mesmo é 0 e 1 respectivamente. No entanto, para números intervalares estes mesmos resultados são obtidos somente em casos particulares. Observe a subtração  $[x] - [x]$ :

$$[\underline{x} - \bar{x}, \bar{x} - \underline{x}] = 0 \Leftrightarrow \underline{x} = \bar{x}.$$

Este problema que surge com as múltiplas ocorrências de variáveis intervalares é denominado de *dependência* e tem como conseqüência principal o alargamento do intervalo resultante, deixando-o menos preciso. Para reduzir a dependência, deve-se inicialmente reorganizar as expressões matemáticas de forma a reduzir o número de vezes que uma mesma variável aparece. Veja como fazer isso através do Exem. 3 e do Exem. 4.

**Exemplo 3 (Tratando a dependência)** Considere a função  $f([x]) = [x]/(1+[x])$ . O valor de  $f$  para o intervalo  $[1, 3]$  pode ser computado como:

a) substituindo  $[1, 3]$  em  $f([x])$ , tem-se

$$f([x]) = \frac{[1, 3]}{1 + [1, 3]} = [0.25, 1.50];$$

b) reorganizando  $f([x])$  obtém-se  $f'([x]) = \frac{1}{1+\frac{1}{[x]}}$ , ou seja, foi extraída a ocorrência extra da variável  $[x]$  mantendo-se uma expressão equivalente a função original. Assim

$$f'([x]) = \frac{1}{1 + \frac{1}{[1,3]}} = [0.50, 0.75].$$

Ambas funções  $f([x])$  e  $f'([x])$  estão corretas. Mas,  $f'([x])$  é mais precisa e portanto deve ser aplicada no lugar de  $f([x])$ .

**Exemplo 4 (Tratando a dependência)** Considere a função  $f([x]) = [x]^2 - [x]$ . O valor de  $f$  para o intervalo  $[x] = [-1, 3]$  pode ser computado como:

a) substituindo  $[-1, 3]$  em  $f([x])$ , tem-se

$$[-1, 3]^2 - [-1, 3] = [0, 9] - [-1, 3] = [-3, 10];$$

b) reorganizando  $f([x])$  obtém-se  $f'([x]) = ([x] - \frac{1}{2})^2 - \frac{1}{4}$ . Substituindo  $[-1, 3]$  em  $f'([x])$ , tem-se

$$([-1, 3] - 0.50)^2 - 0.25 = [-1.50, 2.50]^2 - 0.25 = [-0.25, 6].$$

Novamente, as duas soluções são verdadeiras porém  $f([x])$  produz a solução mais pessimista.

Caso não seja possível eliminar a multiplicidade de ocorrências de variáveis, pelo menos deve-se buscar diminuí-las. Adicionalmente, uma outra forma de tratar a dependência, descrita em Hansen e Walster [50], é redefinir operações aritméticas para os casos de dependência. A subtração dependente é definida por

$$[x] \ominus [y] = [\bar{x} - \bar{y}, \underline{x} - \underline{y}] \quad (2.41)$$

As operações de dependência foram desenvolvidas de forma que seja realmente a operação inversa daquela que remove uma variável dependente. Por exemplo, dados  $[x]$  e  $[y]$ , calcula-se  $[z]$  por  $[z] = [x] * [y]$ . No entanto, é possível que  $[y] \neq [x]/[z]$ , pois  $*$  e  $/$  não são operadores que resolvem dependência. Definido-se  $\oslash$  como divisão dependente,  $[y] = [x] \oslash [z]$  recupera  $[y]$  completamente.

Para maiores informações sobre a construção de funções de inclusão mais precisas e a questão do pessimismo, sugere-se a consulta a Benhamou e Older [10] e Chabert e Jaulin [17] respectivamente.

## 2.3.7 Vetores Intervalares

Em muitas situações, é preciso agrupar mais de um intervalo para representar um parâmetro. A maneira formal é feita via vetores intervalares ou *caixas*, que são descritos a seguir.

**Definição 9** Uma caixa  $[\mathbf{x}]$  é um vetor intervalar em  $\mathbb{IR}^{n_x}$  definido como o produto cartesiano de  $n_x$  intervalos fechados

$$[\underline{\mathbf{x}}, \bar{\mathbf{x}}] = [x_1] \times \dots \times [x_n], \quad (2.42)$$

sendo  $\underline{\mathbf{x}}$  e  $\bar{\mathbf{x}}$  os limites inferior e superior de  $[\mathbf{x}]$ .

Como conseqüência da definição dos vetores intervalares, todas as definições descritas anteriormente sobre intervalos em  $\mathbb{IR}$  são estensíveis a  $\mathbb{IR}^{n_x}$ .

Muitos métodos intervalares utilizam medidas relacionadas aos limites inferiores e superiores de uma caixa como medida de incerteza de seus parâmetros. Esta medida é denominada de *largura* e é definida a seguir.

**Definição 10** Considere uma caixa  $[\mathbf{x}] \in \mathbb{IR}^{n_x}$  não vazia e não composta somente intervalos degenerados, então sua largura  $w([\mathbf{x}]) : \mathbb{IR}^{n_x} \rightarrow \mathbb{R}$  é dada por

$$w([\mathbf{x}]) = \max_{i=\{1, \dots, n_x\}} w([x_i]). \quad (2.43)$$

Ainda, por definição  $w(\emptyset) = -\infty$ .

Como dito anteriormente, um intervalo real representa um número real e sua incerteza. Uma estimativa habitual do número que a caixa representa é o seu *centro*, cuja definição vem a seguir.

**Definição 11** Considere uma caixa não vazia  $[\mathbf{x}] \in \mathbb{IR}^{n_x}$ , então seu centro  $\mathbf{c}([\mathbf{x}]) : \mathbb{IR}^{n_x} \rightarrow \mathbb{R}^{n_x}$  é dado por

$$c_i([\mathbf{x}]) = (\bar{x}_i + \underline{x}_i)/2, \quad i = \{1, \dots, n_x\}. \quad (2.44)$$

Além da largura e do centro pode-se introduzir uma outra medida para caracterizar os vetores intervalares, o *volume*.

**Definição 12** Considere uma caixa não vazia  $[\mathbf{x}] \in \mathbb{IR}^{n_x}$ ,  $n_x \geq 2$ . Então, o volume  $v([\mathbf{x}]) : \mathbb{IR}^{n_x} \rightarrow \mathbb{R}$  é dado por

$$v([\mathbf{x}]) = \prod_{i=\{1, \dots, n_x\}} w([x_i]). \quad (2.45)$$

Por definição, a) se  $x \in \mathbb{R}$ ,  $v([x]) = w([x])$ ; e b)  $v(\emptyset) = -\infty$ .

Se  $w([\mathbf{x}]) \neq 0$ , então existe pelo menos um plano de corte que divide  $[\mathbf{x}]$  em duas outras caixas. É chamado de *plano principal* aquele originado no maior lado da caixa. A operação que divide a caixa é chamada de *bissecção*.

**Definição 13** *A bissecção regular é uma operação que divide uma caixa  $[\mathbf{x}]$  em duas outras simétricas  $[\mathbf{x}]_1$  e  $[\mathbf{x}]_2$ , tais que*

$$[\mathbf{x}]_1 = [x_1] \times \dots \times [x_j, \frac{x_j + \bar{x}_j}{2}] \times \dots \times [x_n], \quad (2.46)$$

$$[\mathbf{x}]_2 = [x_1] \times \dots \times [\frac{x_j + \bar{x}_j}{2}, \bar{x}_j] \times \dots \times [x_n], \quad (2.47)$$

sendo  $j = \min\{i \mid w([x_i]) = w([\mathbf{x}])\}$ .

Neste trabalho, a bissecção é utilizada em todos os algoritmos intervalares. O exemplo a seguir, mostra como ela funciona na prática.

**Exemplo 5 (Operação de bissecção)** *Considere a caixa  $[\mathbf{x}] = [-2, 3] \times [2, 4] \times [0, 3]$ . A operação de bissecção inicia-se com o cálculo da largura da caixa*

$$w([\mathbf{x}]) = w([-2, 3] \times [2, 4] \times [0, 3]) = 5. \quad (2.48)$$

*Finalmente, utilizando o intervalo de maior largura como plano principal de corte a bissecção produz:*

$$\begin{aligned} [\mathbf{x}]_1 &= ([-2, 0.5] \times [2, 4] \times [0, 3]) \\ [\mathbf{x}]_2 &= ([0.5, 3] \times [2, 4] \times [0, 3]) \end{aligned}$$

### 2.3.8 Funções de Inclusão

Segundo Jaulin *et al.* [63], um dos objetivos principais da Análise Intervalar é oferecer, para uma grande classe de funções  $\mathbf{f}$ , *funções de inclusão*  $\mathbf{f}([\mathbf{x}])$  que possam ser avaliadas rapidamente. Em outras palavras, as funções de inclusão são formulações equivalentes, consistentes e que resultem em solução mais rápida que o modelo inicialmente proposto.

**Definição 14** *Considere a função  $\mathbf{f} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_f}$ . A função  $[\mathbf{f}] : \mathbb{IR}^{n_x} \rightarrow \mathbb{IR}^{n_f}$  é uma função de inclusão para  $\mathbf{f}$  se:*

$$\forall [\mathbf{x}] \in \mathbb{IR}^{n_x}, \mathbf{f}([\mathbf{x}]) \subseteq [\mathbf{f}]([\mathbf{x}]) \quad (2.49)$$

A função de inclusão  $[f]$  comporta-se como uma *caixa intervalar* e portanto é possível contornar algumas situações como descontinuidade e não convexidade da função  $f$ , que dificultem o seu manuseio. Analise a Fig. 2.4 e observe como uma função de inclusão mapeia uma caixa  $[x]$  para o espaço das funções. As funções de inclusão podem ser classificadas como *estreita*,

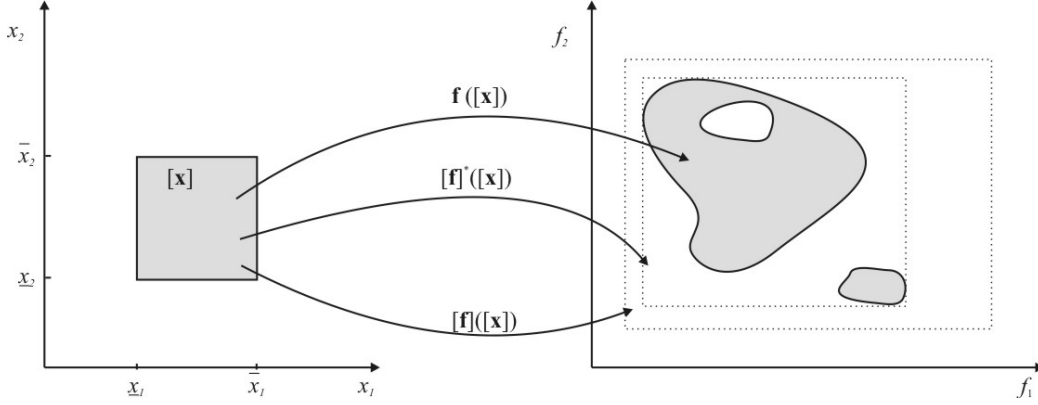


Fig. 2.4: Imagem da caixa  $[x]$  via funções de inclusão.

*convergente, monotônica e mínima.* A função de inclusão é estreita quando

$$\forall \mathbf{x} \in [x], \quad f(\mathbf{x}) = [f](\mathbf{x}). \quad (2.50)$$

Considerando que o índice  $k$  represente o número de bissecções realizadas sobre uma caixa  $[x]$ , a função de inclusão é convergente quando

$$\lim_{k \rightarrow \infty} w([x]_k) = 0 \Rightarrow \lim_{k \rightarrow \infty} w([f]([x]_k)) = 0, \quad (2.51)$$

e monotônica quando

$$[x]_k \subset [x]_{k-1} \Rightarrow [f]([x]_k) \subset [f]([x]_{k-1}). \quad (2.52)$$

Finalmente, a função de inclusão é dita mínima, com notação  $[f]^*([x])$ , quando ela mapear a menor caixa que contém a imagem de  $f$ . A Fig. 2.4 apresenta a ação de  $f$ ,  $[f]^*$  e  $[f]$  sobre a caixa  $[x]$ . Fig. 2.5 e Fig. 2.6 mostram funções de inclusão convergentes, monotônicas e não monotônicas.

As funções de inclusão podem ser construídas a partir de polinômios, de algoritmos e de equações diferenciais. Este trabalho aborda somente as *funções de inclusão naturais*, as quais são descritas a seguir.

**Definição 15** Considere a função  $f : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ , expressada como uma composição finita dos operadores  $+$ ,  $-$ ,  $*$  e  $/$  e de funções elementares (seno, exp, quadrado, ...). A função de inclusão natural pode ser obtida pela substituição de cada operador e função elementar pelo seu modelo intervalar equivalente.

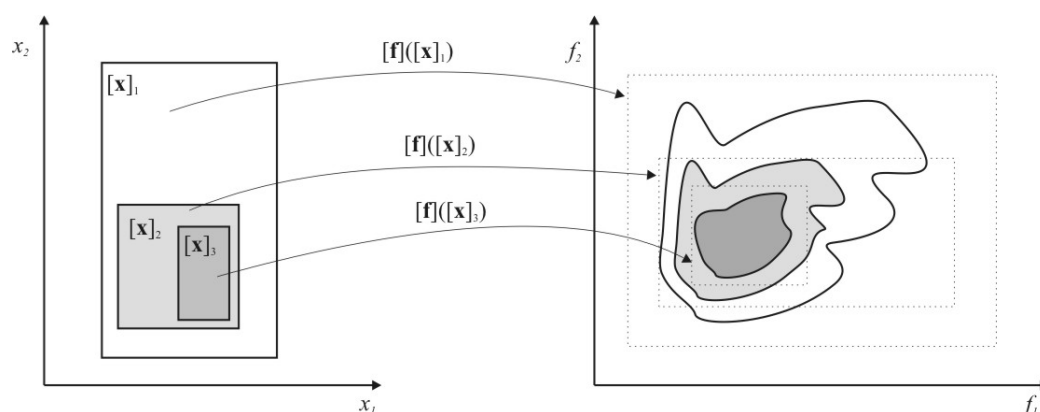


Fig. 2.5: Funções de inclusão monotônica e convergente.

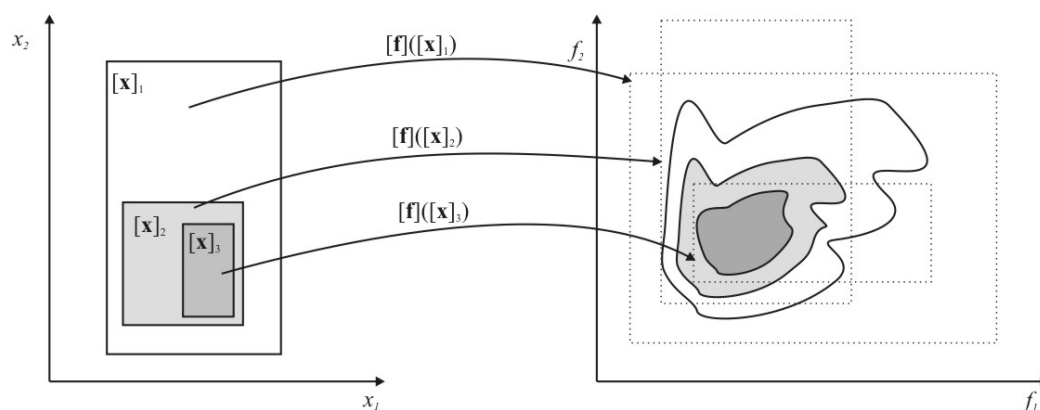


Fig. 2.6: Funções de inclusão não monotônica e convergente.

Em Jaulin *et al.* [63] (pg.30), um teorema sobre as funções de inclusão naturais diz que quando a função natural  $[f]$  envolve apenas operadores e funções elementares contínuos, então  $[f]$  é convergente. Ainda, se cada variável ocorrer no máximo uma vez,  $[f]$  é mínima. A característica de ser mínima está diretamente ligada à precisão da função de inclusão  $[f]$ . Em geral as funções de inclusão naturais não são mínimas por causa, por exemplo, da dependência. Então, em alguns casos, a expressão matemática da função de inclusão pode ser rearranjada de modo a reduzir as ocorrências extras de variáveis.

Os algoritmos de otimização que utilizam intervalos necessitam de informações das funções de inclusão para garantir que reduções no espaço de busca não conduzam à perda do ponto ótimo. Por isso é importante que se construa funções de inclusão mínimas, convergentes e monotônicas.

## 2.3.9 Lógica Booleana Intervalar

A lógica booleana intervalar tem papel fundamental nos métodos intervalares. Por exemplo, ela auxilia em testes de continência entre intervalos. O conjunto booleano intervalar  $\mathbb{IB}$  possui, além dos elementos *verdadeiro* (1) e *falso* (0), as opções *impossível* ( $\emptyset$ ) e *indeterminado* ( $[0, 1]$ ). Pela notação de conjuntos,

$$\mathbb{IB} = \{\emptyset, 0, 1, [0, 1]\}. \quad (2.53)$$

O elemento indefinido  $[0, 1]$  de  $\mathbb{IB}$ , é empregado quando há dúvida no julgamento de alguma expressão. Por outro lado,  $\emptyset$  é utilizado, por exemplo, quando não existe intersecção entre intervalos. Pelas equações acima,  $([0, 1] \vee 1) \wedge ([0, 1] \wedge 1) = 1 \wedge [0, 1] = [0, 1]$ . Da mesma forma que os operadores lógicos, os operadores relacionais também aplicam-se a intervalos. Veja alguns casos nos exemplos que seguem.

**Exemplo 6 (Operadores lógicos nas comparações de intervalos)**

$$\begin{aligned} [x] \leq [y] &\rightarrow \mathbb{B} = 1, && \text{se } \bar{x} \leq \underline{y}. \\ &\rightarrow \mathbb{B} = 0, && \text{se } \underline{x} > \bar{y}. \\ &\rightarrow \mathbb{B} = [0, 1], && \text{se } (\bar{x} > \underline{y} \wedge \underline{x} \leq \bar{y}). \\ [x] = [y] &\rightarrow \mathbb{B} = 1, && \text{se } (\underline{x} = \underline{y} \wedge \bar{x} = \bar{y}), \\ &\rightarrow \mathbb{B} = 0, && \text{caso contrário.} \end{aligned} \quad (2.54)$$

Se operadores como  $\wedge$  e  $\vee$  associam-se a conjuntos e intervalos, então podem ser estendidos para o conjunto  $\mathbb{IB}$ . As expressões que seguem tratam destas definições.

$$[x] \vee [y] = \{x \vee y \mid x \in [x], y \in [y]\}. \quad (2.55)$$

$$[x] \wedge [y] = \{x \wedge y \mid x \in [x], y \in [y]\}. \quad (2.56)$$

$$\neg[x] = \{\neg x \mid x \in [x]\}. \quad (2.57)$$

$$[x] \sqcap [y] = [\max(\underline{x}, \underline{y}), \min(\bar{x}, \bar{y})]. \quad (2.58)$$

$$[x] \sqcup [y] = [\min(\underline{x}, \underline{y}), \max(\bar{x}, \bar{y})]. \quad (2.59)$$

Outros relacionais podem ser obtidos por analogia. Veja o Exem. 7.

**Exemplo 7 (Valores das expressões booleanas)**

$$[-1, 2] \leq [2, 3] \rightarrow \mathbb{B} = 1$$

$$[-1, 2] \geq [4, 6] \rightarrow \mathbb{B} = 0$$

$$[1, 2] \geq [-1, 0] \rightarrow \mathbb{B} = 1$$

$$[-1, 2] \in [0, 6] \rightarrow \mathbb{B} = [0, 1]$$

$$[-1, 2] \cap [3, 5] \rightarrow \mathbb{B} = \emptyset$$

Os relacionais descritos acima foram implementados em um formato estendido para comparação de caixas intervalares. Expressões booleanas, intervalares ou não, são empregadas em diversas situações, como por exemplo para direcionar passos de métodos e algoritmos. Entre os usos mais freqüentes na Análise Intervalar está a aplicação nos *testes de inclusão*.

### 2.3.10 Teste de Inclusão

Um *teste* é uma função  $t(\mathbf{x}) : \mathbb{R}^{n_x} \rightarrow \mathbb{B}$ . Similarmente, um *teste de inclusão* é uma função de  $[t](\mathbf{x}) : \mathbb{IR}^{n_x} \rightarrow \mathbb{IB}$  tal que para qualquer  $\mathbf{x} \in \mathbb{IR}^{n_x}$ ,

$$\begin{aligned} [t](\mathbf{x}) = 1 &\Rightarrow \forall x \in \mathbf{x}, t(\mathbf{x}) = 1, \\ [t](\mathbf{x}) = 0 &\Rightarrow \forall x \in \mathbf{x}, t(\mathbf{x}) = 0. \end{aligned}$$

O teste de inclusão  $[t]$  é *estrito* se  $[t](\mathbf{x}) \neq [0, 1]$  para qualquer  $\mathbf{x} \in \mathbb{R}^{n_x}$ , e *mínimo* se

$$\forall \mathbf{x} \in \mathbb{IR}^{n_x}, [t](\mathbf{x}) = \{t(\mathbf{x}) \mid \mathbf{x} \in \mathbf{x}\} \quad (2.60)$$

Um teste de inclusão mínimo é necessariamente estrito. O Exem. 8 a seguir apresenta uma função para teste de inclusão.

**Exemplo 8 (Teste de inclusão)** *Considere o teste  $t(\mathbf{x}) = x_1 + x_2$ ,  $0.5 \leq x_1^2 + x_2^2 \leq 1$ . Um teste de inclusão  $[t](\mathbf{x})$  para este problema pode ser escrito como*

$$[t](\mathbf{x}) : \begin{cases} 1, & \text{se } [x_1]^2 + [x_2]^2 \subseteq [0.5, 1]; \\ 0, & \text{se } ([x_1]^2 + [x_2]^2) \cap [0.5, 1] = \emptyset; \\ [0, 1], & \text{caso contrário.} \end{cases} \quad (2.61)$$

Um teste de inclusão pode ser formado pela composição de vários outros testes de inclusão utilizando-se de operadores booleanos para efetuar a integração.

### 2.3.11 Subpavimentos

Substituir uma função  $\mathbf{f}(\mathbf{x})$  por uma correspondente função de inclusão  $[\mathbf{f}](\mathbf{x})$  facilita a realização de cálculos sobre ela, mas perde-se precisão devido à porção extra de espaço não vinculada à função, que surge da computação de intervalos. Este problema pode ser amenizado se os argumentos intervalares forem decompostos em caixas menores. Jaulin *et al.* [63] definem a caixa inicial  $\mathbf{x}$  como *pavimento* e caixas que surgem da decomposição de *subpavimentos*  $\mathbf{x}_1$  e  $\mathbf{x}_2$ . O método utilizado para dividir as caixas é a

bissecção. Neste trabalho, a bissecção é realizada conforme (2.46) e (2.47) produzindo duas caixas simétricas e conseqüentemente com mesmo volume, por isso os subpavimentos são ditos *regulares*.

Os métodos que empregam subpavimentos executam bissecções continuamente até que algum critério de parada seja satisfeito. Ao longo das etapas, os subpavimentos são convenientemente armazenados e organizados em estruturas de listas encadeadas do tipo *filas* FIFO (first in first out) e *pilhas* LIFO (last in first out). Os métodos desenvolvidos aqui fazem uso destas listas e por isso uma pequena revisão das operações em uma FIFO e em uma LIFO são apresentadas no Exem. 9 e no Exem. 10.

**Exemplo 9 (Operações em fila FIFO)** *Considere a fila  $\mathbf{Q}_{[\mathbf{x}]}$  de parâmetros  $[\mathbf{x}]$  e o banco de dados com as caixas  $[\mathbf{x}]_1$  e  $[\mathbf{x}]_2$ .*

- 1 *inicializar  $\mathbf{Q}_{[\mathbf{x}]}$  com  $\emptyset$*   $\Leftrightarrow \mathbf{Q}_{[\mathbf{x}]} = \emptyset$
- 2 *insira  $[\mathbf{x}]_1$  em  $\mathbf{Q}_{[\mathbf{x}]}$*   $\Leftrightarrow \mathbf{Q}_{[\mathbf{x}]} = \{[\mathbf{x}]_1\}$
- 3 *insira  $[\mathbf{x}]_2$  em  $\mathbf{Q}_{[\mathbf{x}]}$*   $\Leftrightarrow \mathbf{Q}_{[\mathbf{x}]} = \{[\mathbf{x}]_1, [\mathbf{x}]_2\}$
- 4 *retire elemento de  $\mathbf{Q}_{[\mathbf{x}]}$*   $\Leftrightarrow \mathbf{Q}_{[\mathbf{x}]} = \{[\mathbf{x}]_2\}$

**Exemplo 10 (Operações em pilha LIFO)** *Considere a pilha  $\mathbf{S}_{[\mathbf{x}]}$  de parâmetros  $[\mathbf{x}]$  e o banco de dados com as caixas  $[\mathbf{x}]_1$  e  $[\mathbf{x}]_2$ .*

- 1 *inicializar  $\mathbf{S}_{[\mathbf{x}]}$  com  $\emptyset$*   $\Leftrightarrow \mathbf{S}_{[\mathbf{x}]} = \emptyset$
- 2 *empilhe  $[\mathbf{x}]_1$  em  $\mathbf{S}_{[\mathbf{x}]}$*   $\Leftrightarrow \mathbf{S}_{[\mathbf{x}]} = \{[\mathbf{x}]_1\}$
- 3 *empilhe  $[\mathbf{x}]_2$  em  $\mathbf{S}_{[\mathbf{x}]}$*   $\Leftrightarrow \mathbf{S}_{[\mathbf{x}]} = \{[\mathbf{x}]_1, [\mathbf{x}]_2\}$
- 4 *desempilhe de  $\mathbf{S}_{[\mathbf{x}]}$*   $\Leftrightarrow \mathbf{S}_{[\mathbf{x}]} = \{[\mathbf{x}]_1\}$

A utilização de listas ou pilhas para armazenar os subpavimentos regulares traz vantagens no controle e na organização dos elementos pertencentes a elas. Para manter a regularidade dos subpavimentos, o método intervalar deve garantir que todas as caixas numa dada iteração passem pelo mesmo número de bissecções  $n_{bis}$ . Uma *rodada de bissecção*  $r_{bis}$  se completa quando todos os elementos da estrutura de dados sofrem uma bissecção e o resultado sempre será composto por caixas simétricas com metade do volume das caixas da rodada anterior. Ao final de  $r_{bis}$  rodadas de bissecção ter-se-á  $2^{r_{bis}}$  subpavimentos, se não houverem exclusões.

### 2.3.12 SIVIA

O algoritmo SIVIA (Set Inversion Via Interval Analysis), proposto por Jaulin [63] (pg.56), é um método que classifica subpavimentos quanto a pertinência ou não em funções de inclusão  $[\mathbf{f}]$ . Em poucas palavras, SIVIA busca pelo

domínio, dada a imagem da função  $[y]$ . SIVIA tem como entrada o espaço de busca  $[x]$ , o intervalo-imagem  $[y]$  que deseja encontrar o domínio, e o parâmetro de precisão  $\varepsilon$  para limitar a discretização dos subpavimentos de  $[x]$ . Com relação ao processo, SIVIA bissecta  $[x]$  até classificar todos os seus subpavimentos. Usualmente, SIVIA produz como resultado três listas encadeadas: a) uma lista contendo as *caixas solução*  $[x]_s$  que satisfazem  $[f]([x]_s) \subseteq [y]$ ; b) uma lista formada pelas *caixas não solução*  $[x]_n$ , tais que  $[f]([x]_n) \cap [y] = \emptyset$ ; e c) uma outra lista constituída pelas *caixas de fronteira*  $[x]_f$  que não foram classificadas quanto ao critério (a) e nem quanto ao critério (b). SIVIA foi amplamente utilizado neste trabalho para eliminar porções não viáveis do espaço de busca, usando nestes casos, funções de inclusão para tratar as restrições do problema.

### 2.3.13 Inequações Intervalares

Após abordar vários aspectos da matemática para intervalos importantes para o tratamento de funções objetivo e de restrição nos algoritmos de otimização intervalares, resta apresentar, mesmo que sucintamente, as *inequações intervalares*.

Uma inequação que contenha variáveis e/ou coeficientes intervalares é denominada de inequação intervalar. Por exemplo, considere o parâmetro  $[x] \in \mathbb{IR}^{n_x}$ , uma função  $g([x]) : \mathbb{IR}^{n_x} \rightarrow \mathbb{IR}$  e o intervalo  $[y] \in \mathbb{IR}$ . Uma inequação intervalar pode ter a forma

$$g([x]) \leq [y]. \quad (2.62)$$

O conjunto solução para a inequação (2.62) pode ser dado por

$$[X]^* = \{[x]^* \in \mathbb{IR}^{n_x} \mid \bar{g}([x]^*) \leq \underline{y}\}. \quad (2.63)$$

Claramente,  $\forall x \in [x]^*$  é a solução real de (2.62). Agora considere um sistema de  $n_g$  inequações intervalares formado pela composição de inequações  $g_i([x]) \leq [y]_i$ ,  $i = \{1, \dots, n_g\}$ . O conjunto solução pode ser reescrito como

$$[X]^* = \bigcap_{i=\{1, \dots, n_g\}} \{[x]^* \in \mathbb{IR}^{n_x} \mid \bar{g}_i([x]^*) \leq \underline{y}_i\}. \quad (2.64)$$

Como exemplo de aplicação, as inequações intervalares podem ser utilizadas no auxílio à resolução de sistemas de equações intervalares. Veja o Exem.11. A caracterização de um sistema de equações intervalares é intuitiva e por isso a definição formal não será realizada. A Fig. 2.7 apresenta os intervalos solução.

**Exemplo 11 (Sistema de equações intervalares)** *Considere o sistema proposto por Hansen e Walster [50] (pg.87):*

$$\begin{cases} [2, 3] * [x_1] + [0, 1] * [x_2] = [0, 120] \\ [1, 2] * [x_1] + [2, 3] * [x_2] = [60, 240] \end{cases}$$

*Para resolver este problema é necessário analisar separadamente cada quadrante. Se por exemplo  $\mathbf{x}$  estiver no primeiro quadrante, os limites inferiores de  $[x_1]$  e  $[x_2]$  são maiores ou iguais a 0 e portanto o sistema pode ser reescrito como:*

$$\begin{cases} [2 * [x_1], 3 * [x_1] + [x_2]] = [0, 120] \\ [[x_1] + 2 * [x_2], [x_1] + 3 * [x_2]] = [60, 240] \end{cases}$$

*Os valores dos limites inferiores do lado esquerdo do sistema não podem ultrapassar os limites superiores do lado direito. Por outro lado, os limites superiores do lado esquerdo não podem ser menores que os limites inferiores do lado direito. Matematicamente:*

$$\begin{cases} 2 * [x_1] \leq 120 & 3 * [x_1] + [x_2] \geq 0 \\ [x_1] + 2 * [x_2] \leq 240 & [x_1] + 3 * [x_2] \geq 60 \end{cases}$$

*Resolvendo as inequações estabelece-se os limites para  $[x_1] = [-120, 90]$  e  $[x_2] = [-60, 240]$ . Observa-se que, se os intervalos solução forem inseridos nas equações, a resposta é mais larga que o lado direito do sistema.*

Definições e métodos para resolver equações intervalares podem ser encontrados em Vaccaro [120]. Adicionalmente, os interessados em sistemas de equações, podem ser também suportados por Hansen e Walster [50].

## 2.4 Otimização Multi-objetivo Intervalar

Dentre os primeiros indícios da otimização utilizando intervalos destaca-se um relatório técnico da Universidade de Stanford em 1962 descrito por Moore. Neste estudo, os intervalos eram reduzidos conforme a monotonicidade das funções intervalares e quando eram satisfeitas algumas condições para aplicação do método de otimização de Newton. Além disso a subdivisão de intervalos já estava prevista. Mais tarde, o conteúdo deste relatório foi inserido no livro [85], também de Moore em 1966, considerado por muitos autores como a primeira referência de Análise Intervalar. Em 1974, Skelboe [107] desenvolveu um método para computar as formulações propostas por Moore e deixar a subdivisão de intervalos mais rápida. Atualmente, este método é conhecido como o algoritmo de Moore-Skelboe. Moore [86],

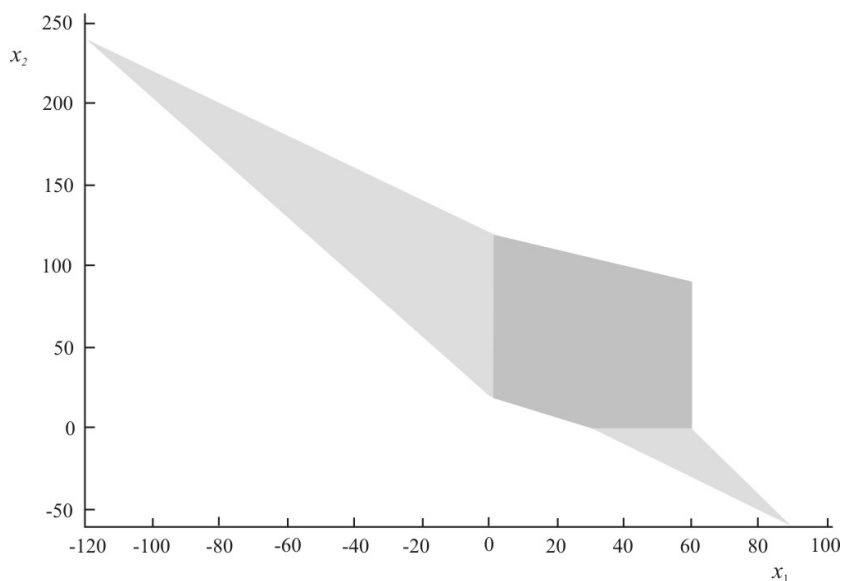


Fig. 2.7: Soluções de equações intervalares com duas equações e duas variáveis. A parte acinzentada mais escura representa o intervalo correspondente à análise do primeiro quadrante. A parte em destaque restante corresponde ao resultado completo do sistema.

influenciado pelo trabalho de Skelboe e pela versão do método de Newton elaborada Krawczyk desenvolveu um algoritmo para otimização global. O método de Krawczyk e outras variantes do método de Newton intervalar podem ser encontradas em Alefeld [1]. No entanto, o emprego da terminologia *otimização global* utilizando-se de algoritmos intervalares foi primeiramente realizada por Hansen em [49], por Skelboe [108] e por Ichida e Fujii [60], sendo todos os trabalhos no ano de 1979.

Mesmo com início da década de 1960, a otimização intervalar foi abordada em relativamente poucos trabalhos quando se compara com número de publicações com outras classes de métodos, como a classe dos métodos evolucionários. Os motivos da não preferência pelos métodos intervalares normalmente envolvem questões de custo computacional, mesmo tendo conhecimento que a garantia de convergência para a região de otimalidade pode compensar esse suposto esforço. Hansen e Walster [50] defendem o uso dos métodos intervalares e são referência quando o assunto é otimização intervalar. Os métodos intervalares aplicados à otimização têm usualmente a mesma filosofia de busca: eliminar regiões onde seguramente a solução ou o conjunto solução não está. Assim, as regiões restantes certamente contêm o(s) ótimo(s) do problema. Por outro lado, os algoritmos intervalares também podem ser estendidos de métodos de otimização que utilizam variáveis,

constantes e funções definidas em  $\mathbb{R}$ . Nestes casos, os algoritmos equivalentes podem ser obtidos através da substituição dos termos  $\mathbb{R}$  por intervalos  $\mathbb{IR}$  que os contêm e da adequação de métodos e das operações aritméticas para intervalos.

Ichida e Fujii [61], em 1990, também estiveram entre os primeiros pesquisadores a utilizar-se de métodos intervalares para resolver problemas multi-objetivos. Neste estudo, as relações clássicas de dominância foram contextualizadas para realizar comparações entre intervalos; intervalos dominados significam ausência de solução e permissão para descartá-lo. Comparações desta natureza são freqüentemente encontradas nos algoritmos intervalares multi-objetivo atuais. No entanto, a comparação e posteriormente escolha entre dois intervalos depende da opinião do projetista. Por exemplo, um projetista pode ser considerado “otimista” quando ele declara preferência a um intervalo segundo uma dada vantagem posicional entre o intervalo preferencial e o outro intervalo. Por outro lado, um projetista é classificado como “pessimista” quando a preferência por intervalo a outro é clara posicionalmente; todos os pontos contidos no intervalo preferido dominam todos os pontos contidos no outro intervalo. Os pontos de vista otimista e pessimista são claramente explicados por Mahato e Bhunia [78]. A Fig.2.8 ilustra uma configuração onde a estratégia pessimista resulta em menor exclusão de intervalos. Na figura, assumamos que  $[\mathbf{x}] \in \mathbb{IR}^{n_x}$  represente qualquer uma das

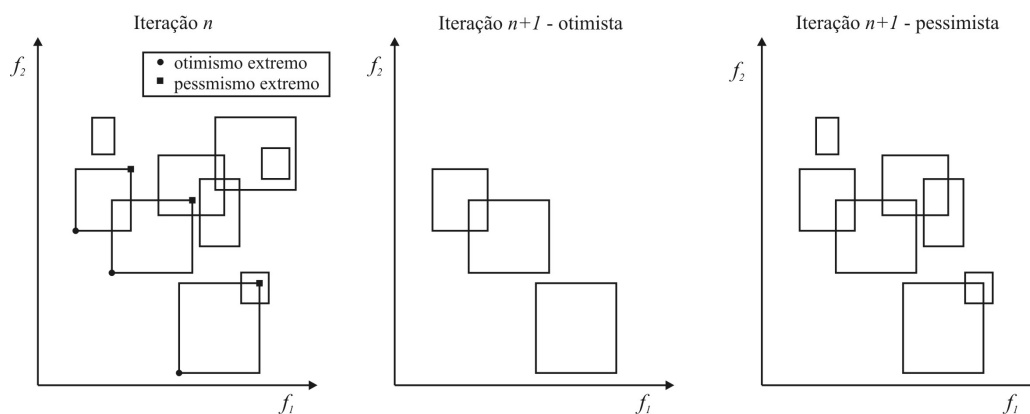


Fig. 2.8: O gráfico à esquerda representa a iteração  $n$  onde 9 intervalos devem ser comparados para compor a iteração  $n + 1$ . A fronteira de Pareto deveria ser formada pelos pontos “•”, caso fosse considerado o otimismo extremo, ou pelos pontos “■”, no caso de pessimismo extremo. Os gráficos no centro e à direita mostram os intervalos não dominados segundo as visões otimista e pessimista respectivamente.

caixas. Definiu-se por *otimismo extremo* utilizar o ponto  $\underline{\mathbf{x}}$  como referên-

cia para dominância, e por *pessimismo extremo*, o ponto  $\bar{\mathbf{x}}$ . Otimismo e/ou pessimismo não extremos definem usualmente valores estimados da solução como os pontos de referência, como por exemplo o centro de cada caixa. Existem outras formas de se comparar intervalos. Dentre os trabalhos que discutem a comparação de intervalos podem ser citados Casado *et al.* [16], Csallener *et al.* [23] e Sengupta e Pal [105]. Ainda pela Fig. 2.8, nota-se que a estratégia pessimista é conservadora. Como consequência, há garantia de encontrar todas as soluções do problema.

Ichida e Fujii [61] desenvolveram versões intervalares para os métodos multi-objetivo *soma ponderada* e *método min max*. O uso de derivadas foi empregado para direcionar a busca. Ruetsch [100] propôs um algoritmo híbrido onde a busca é global mas guiado por direções apontadas pelo gradiente. Os testes foram realizados utilizando-se de funções objetivo convexas, não convexas e com fronteira de Pareto multi-modal. Barichard e Hao [6] desenvolveram um algoritmo multi-objetivo intervalar baseado em populações. Diferentemente dos outros algoritmos, a bissecção ocorre no espaço dos objetivos. Como consequência, as bissecções causam redundâncias no espaço de busca que são eliminadas por técnicas ICP (Interval Constraint Propagation), consulte Benhamou e Older [10] e Jaulin [62] para maiores informações sobre técnicas ICP.

## 2.5 Considerações Finais

Os problemas de otimização reais são, em sua maioria, multi-objetivo. E, a existência de critérios conflitantes motiva os responsáveis pelo sistema de otimização a indicar suas preferências. Os métodos multi-objetivo citados foram classificados segundo a ordem de se articular as preferências. No entanto, outras classificações são possíveis, como por exemplo os métodos podem depender ou não de cálculo de derivadas e ainda, os métodos podem ser determinísticos ou estocásticos. O fato é que não existe nenhum algoritmo que seja reconhecidamente melhor. De maneira geral, os problemas de otimização possuem suas particularidades, as quais definem os algoritmos a serem utilizados. Neste trabalho, a otimização multi-objetivo apresentada até agora sofre mudanças com a inserção do parâmetro de incerteza. Logo, a definição do problema multi-objetivo é alterada para a nova realidade, a otimização robusta multi-objetivo.

Em muitos casos, os algoritmos evolucionários mantêm uma estrutura básica de funcionamento, diferenciando-se uns dos outros por detalhes de implementação em métodos como os de seleção e técnica de nicho.

A representação de um parâmetro  $\mathbf{x} \in \mathbb{R}^{n_x}$  por seu correspondente in-

tervalo  $[\mathbf{x}] \in \mathbb{IR}^{n_x}$  traz mudanças profundas nos métodos empregados para tratamento do referido parâmetro. Por isso, julgou-se necessário realizar a apresentação de definições básicas sobre alguns aspectos da álgebra intervalar como operações aritméticas, funções e inequações que contemplem o tipo de dado intervalar. Importante registrar que a álgebra intervalar é usualmente classificada como conservativa ou pessimista, propagando todas as instâncias dos intervalos que participam do processo. Exatamente por esse motivo escolheu-se os métodos intervalares para suportar a construção de algoritmos para otimização robusta: a propagação da incerteza deve ser realizada em problemas onde a resposta robusta é única realmente desejada, observando é claro a precisão oferecida pelos intervalos solução.

Considerando que os métodos intervalares podem substituir métodos pontuais em diversos problemas de engenharia e ainda proporcionar garantia de envolver a solução do problema, surge a questão do porque de terem sido freqüentemente utilizados. Dentre as respostas, o ponto mais levantado está relacionado ao tempo computacional. Realmente, em grande parte dos problemas a aritmética real é muito mais rápida e precisa que a intervalar. No entanto, há situações onde o tempo computacional é equivalente e, em outros momentos, até mesmo mais rápida. Como a velocidade dos processadores está cada vez maior e com a construção de compiladores próprios para resolver problemas via Análise Intervalar, provavelmente os algoritmos apresentados neste capítulo tenham mais uso.

### 3. ALGORITMOS PARA OTIMIZAÇÃO ROBUSTA MULTI-OBJETIVO

O termo “robusto” tem sido empregado em diferentes contextos. Por exemplo, Kitano [71] define robustez como a manutenção de específicas funcionalidades de um sistema contra perturbações do ambiente. Em engenharia de software, De Vale e Koopman [35] usaram robustez como sinônimo de tolerância a erros em software. Para Du e Chen [38], em engenharia de projeto, o vocábulo robustez está ligado à escolha do melhor modelo que relaciona compromissos com atributos tipo média e a variância, por exemplo. Nas aplicações em engenharia, robustez significa confiabilidade do sistema quando, eventualmente, falhas ocorrem ou na presença de incertezas. Mesmo quando o significado muda, intuitivamente, a palavra robustez faz lembrar características como força, resistência, flexibilidade e adaptabilidade.

No contexto de otimização, Parkinson *et al.* [93] apresentaram uma metodologia robusta baseada no *pior caso de projeto*. Similarmente, Kouvelis e Yu [75] definiram o problema da otimização robusta como a minimização do máximo desvio de uma solução candidata frente às incertezas aos quais o sistema está sujeito. Outras definições de robustez categorizam as soluções robustas. Neste sentido, Roy [98] definiu *análise de robustez* como o processo de encontrar *conclusões robustas* que podem ser *perfeitamente robustas*, *aproximadamente robustas* e *pseudo-robustas*. Em [36], Dias and Clímaco classificaram as conclusões robustas em *robusto absoluto*, *robusto binário relativo* e *robusto unário relativo*. Classificações como as feitas acima são úteis principalmente quando as incertezas são difíceis de computar e/ou quando não é possível prever aproximadamente sua intensidade. Isto motivou vários pesquisadores a concentrar sua atenção em métodos probabilísticos. No ponto de vista deste trabalho com relação à otimização robusta, as soluções robustas são aquelas que têm o melhor desempenho para o pior caso da interferência de incerteza. Desse modo, observando dado grau de precisão, uma solução é declarada robusta ou não.

Na Seção 2.1, a otimização multi-objetivo apresenta-se como uma área de pesquisa madura, onde diversos métodos de otimização estão disponíveis. Por outro lado, o modelo de otimização robusta multi-objetivo tem um longo caminho pela frente. São várias as questões a considerar para o desenvolvi-

mento dos algoritmos. Primeiramente, a filosofia robusta a ser adotada. Por exemplo, Calafiore e Campi [14] compararam duas filosofias: a) o *pior caso de projeto*, onde a influência das incertezas é maximizada; e b) a *aproximação probabilística*, onde o termo robusto é definido por um número fixo de amostras. Similarmente, Soares *et al.* [110] apresentaram uma aproximação probabilística para o pior caso de projeto e utilizaram o modelo para otimizar um equipamento SMES (Superconducting Magnetic Energy Storage System). Em segundo lugar, a natureza das incertezas e a maneira de inseri-las no modelo matemático também devem ser levadas em consideração no modelo robusto. Para Beyer e Sendhoff [12], as incertezas estão nas condições operacionais, no espaço dos parâmetros, na medição da saída do sistema e na construção do modelo. Quanto a inserção nos modelos robustos, Jin e Branke [65] apresentam expressões para funções objetivos que incluem ruído, erros por aproximações devido arredondamento, erros na medição e as incertezas em sistemas dinâmicos. Finalmente, Beyer e Sendhoff [12] apresenta três caminhos diferentes para modelar incertezas: a) deterministicamente; b) probabilisticamente; e c) possibilisticamente. Claramente, a otimização robusta pertence a um grupo diferente da otimização tradicional. No entanto, alguns trabalhos trataram os problemas de otimização robusta com métodos que são extensões de versões não robustas de algoritmos evolucionários, determinísticos e híbridos, por exemplo Deb e Gupta [33] e Soares *et al.* [110]. Nesta Tese, decidiu-se por desenvolver métodos construídos especificamente para otimização multi-objetivo robusta, segundo um modelo min max, onde as incertezas ocorrem em qualquer parte do modelo robusto, sendo a computação de incertezas realizada utilizando-se de técnicas inter-valares.

### 3.1 Breve Histórico da Otimização Robusta

Entre os primeiros trabalhos onde se reconhecem os indícios de busca de robustez citam-se Fisher [41] em 1951 que desenvolveu um experimento estatístico para melhorar a produção agrícola. Em outro estudo, Dantzig [26] em 1955 formulou problemas onde as incertezas do sistema dependem dos valores de demanda obtidos em etapa inicial do processo de otimização. Em 1959, Charnes e Cooper [18] apresentaram o modelo probabilístico para otimização robusta denominado *Chance-constrained programming*. No entanto, muitos autores convergem em indicar que a teoria da otimização robusta iniciou com Taguchi, também no final dos anos 50, cujo trabalho teve como foco o desenvolvimento de produtos de alta qualidade mesmo considerando a possível influência de fontes de ruído. Taguchi [118] apresentou uma metodologia

em três estágios: a) projeto de sistema; b) projeto de parâmetro; e c) projeto de tolerância. Sucintamente, os requisitos de qualidade são definidos em (a), segundo as tecnologias vigentes com comprovada maturidade industrial, em um formato de uma *função de perda de qualidade*. Em (b), o objetivo é estabelecer níveis de otimalidade dos parâmetros de projeto independente do ambiente em que está imerso ou das condições de uso. Em outras palavras, os parâmetros devem ser “robustos” aos fatores de ruído. Em (c), são definidas estratégias para reduzir a variação de desempenho ou aumentar a tolerância do produto quando este estiver submetido às possíveis fontes de incerteza. Existem diversos trabalhos que utilizaram a metodologia robusta de Taguchi em diferentes áreas como economia, eletrônica e na indústria automotiva, por exemplo.

Os conceitos definidos por Taguchi relativos à interferência de alguma forma de ruído no desempenho dos parâmetros de projeto influenciaram o desenvolvimento da otimização robusta atual. Diferentemente de outros pesquisadores, Taguchi tratou ruído como um parâmetro à parte. Ben-Tal e Nemirovski também simularam a ação das incertezas como um parâmetro independente, nos seus trabalhos sobre otimização convexa robusta [7] e [8]. Em sistemas dinâmicos, El Ghaoui *et al.* [40] e El Ghaoui e Calafiori [39] estiveram entre os precursores. Em outra direção, Kouvelis e Yu [75] e Bertsimas e Sim [11] estenderam a metodologia robusta para sistemas discretos. Técnicas de análise de sensibilidade têm sido empregadas após procedimentos tradicionais de otimização para simular o ambiente de incertezas e encontrar soluções robustas. Como exemplos Mareschal [79], e Dias e Clímaco [36].

## 3.2 Conceitos e Notações

### 3.2.1 Comparação de vetores reais e intervalares

As metodologias multi-objetivo freqüentemente definem relações de dominância baseadas em avaliações de funções objetivo para comparar, classificar, selecionar e rejeitar soluções candidatas em seus procedimentos. Em otimização robusta, realizar as tarefas acima tornam-se mais complicadas pois o valor da função objetivo varia dependendo da intensidade das incertezas. Por isso, para afirmar que uma solução domina outra, é necessário computar todas as instâncias das funções objetivo sobre as soluções que se deseje confrontar. Decidiu-se neste trabalho, substituir as relações de dominância por operadores descritos pelas definições a seguir.

**Definição 16** Considere  $\mathbf{u} \in \mathbb{R}^{n_n}$ ,  $\mathbf{v} \in \mathbb{R}^{n_n}$ ,  $[\mathbf{t}] \in \mathbb{I}\mathbb{R}^{n_n}$  e  $[\mathbf{z}] \in \mathbb{I}\mathbb{R}^{n_n}$ . O

operador de comparação  $\preceq$  é redefinido como segue

$$\mathbf{u} \preceq \mathbf{v} \Leftrightarrow \mathbf{u} \leq \mathbf{v}, \mathbf{u} \neq \mathbf{v}; \quad (3.1)$$

$$\mathbf{u} \preceq [\mathbf{t}] \Leftrightarrow \mathbf{u} \preceq \underline{\mathbf{t}}; \quad (3.2)$$

$$[\mathbf{z}] \preceq [\mathbf{t}] \Leftrightarrow \bar{\mathbf{z}} \preceq \underline{\mathbf{t}}. \quad (3.3)$$

O operador de comparação para o caso estrito  $\prec$  é expresso como

$$\mathbf{u} \prec \mathbf{v} \Leftrightarrow \mathbf{u} < \mathbf{v}. \quad (3.4)$$

Por analogia estende-se os operadores  $\succeq$  e  $\succ$ .

A expressão  $\mathbf{u} \preceq \mathbf{v}$  deve ser lida como  $\mathbf{u}$  é menor que  $\mathbf{v}$  ou  $\mathbf{v}$  é maior que  $\mathbf{u}$ , adicionando-se *estritamente* quando for o caso. Similar leitura também se aplica às demais variações destes operadores.

Empregando os operadores de comparação acima, os pontos internos a conjuntos reais podem ser classificados nos subconjuntos dos *menores vetores* ou dos *maiores vetores*. Essa distinção é importante nos métodos de otimização multi-objetivo. As definições formais destes conjuntos são apresentadas a seguir.

**Definição 17** Considere o conjunto de vetores  $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m\}$ ,  $\mathbf{U} \subseteq \mathbb{R}^{n_n}$ . Os menores vetores de  $\mathbf{U}$  definem o conjunto  $\Upsilon_{\preceq}$  descrito como

$$\Upsilon_{\preceq}(\mathbf{U}) = \{\mathbf{u} \in \mathbf{U} \mid \nexists \mathbf{u}' \in \mathbf{U}, \mathbf{u}' \preceq \mathbf{u}\}. \quad (3.5)$$

Analogamente, os maiores vetores de  $\mathbf{U}$  formam o conjunto  $\Upsilon_{\succeq}$  expresso por

$$\Upsilon_{\succeq}(\mathbf{U}) = \{\mathbf{u} \in \mathbf{U} \mid \nexists \mathbf{u}' \in \mathbf{U}, \mathbf{u} \preceq \mathbf{u}'\}. \quad (3.6)$$

Agora, utilizando os conceitos de menores e maiores vetores de um conjunto de números reais é possível definir o ponto ideal para maximização a ser usado como referência para identificar o pior caso de uma variável de projeto segundo a presença das incertezas.

**Definição 18** Considere o conjunto de vetores  $\mathbf{U} = \{\mathbf{x}, \mathbf{y}, \dots, \mathbf{z}\} \subset \mathbb{R}^{n_n}$ . O ponto ideal (ou utópico) para maximização  $\mathbf{u}^{\max} \in \mathbb{R}^{n_n}$  do conjunto  $\mathbf{U}$  é

$$u_i^{\max}(\mathbf{U}) = \max_i \{x_i, y_i, \dots, z_i\}, \quad i = 1, \dots, n_n. \quad (3.7)$$

O ponto ideal para maximização pode não pertencer ao conjunto dos maiores vetores de um conjunto. O Teo.1 trata essa questão. Esse fato traz conseqüências diretas na formulação de otimização robusta e nos algoritmos apresentados nas próximas seções, pois a robustez de uma dada solução candidata é medida utilizando-se  $\mathbf{u}^{\max}$ , mesmo sabendo-se da possibilidade de  $\mathbf{u}^{\max}$  não pertencer à imagem das funções objetivo.

**Teorema 1** Considere  $\mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^{n_x}$ ,  $\mathbf{f}(\mathbf{x}) : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_f}$  e  $\mathbf{U} = \{\mathbf{u} \in \mathbb{R}^{n_f} \mid \forall \mathbf{x} \in \mathbf{X}, \mathbf{u} = \mathbf{f}(\mathbf{x})\}$ ,  $\mathbf{U} \neq \emptyset$ . Se  $\Upsilon_{\succeq}(\mathbf{U}) \neq \{\mathbf{u}^{\max}(\mathbf{U})\}$

$$\mathbf{u}^{\max}(\mathbf{U}) \notin \mathbf{U}. \quad \blacksquare \quad (3.8)$$

**Prova:** Assuma  $\mathbf{u}^{\max}(\mathbf{U}) \in \Upsilon_{\succeq}(\mathbf{U})$ . Usando (3.6) e (3.7) conclui-se que  $\Upsilon_{\succeq}(\mathbf{U})$  é o conjunto unitário  $\Upsilon_{\succeq}(\mathbf{U}) = \{\mathbf{u}^{\max}(\mathbf{U})\}$ . Por definição, usando apenas (3.7) conclui-se que  $\forall \mathbf{u} \in \mathbf{U}$ ,  $\mathbf{u}^{\max}(\mathbf{U}) \succeq \mathbf{u}$ . Logo, se  $\mathbf{u}^{\max}(\mathbf{U}) \notin \Upsilon_{\succeq}(\mathbf{U})$  então  $\mathbf{u}^{\max}(\mathbf{U}) \notin \mathbf{U}$ .

### 3.3 Problema Multi-Objetivo Robusto

Considere as variáveis de projeto  $\mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^{n_x}$  e o parâmetro de incerteza  $\mathbf{p} \in \mathbf{P} \subseteq \mathbb{R}^{n_p}$ . Definindo-se o vetor de funções objetivo por  $\mathbf{f}(\mathbf{x}, \mathbf{p}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_f}$  e o vetor de funções de restrição por  $\mathbf{g}(\mathbf{x}, \mathbf{p}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_g}$ , o problema de minimização robusta multi-objetivo pode ser escrito como

$$\begin{aligned} \min_{\mathbf{x} \in \mathbf{X}} \quad & \max_{\mathbf{p} \in \mathbf{P}} \mathbf{f}(\mathbf{x}, \mathbf{p}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{x}, \mathbf{p}) \leq \mathbf{0}. \end{aligned} \quad (3.9)$$

Definindo o espaço viável por

$$\mathbf{X}' = \{\mathbf{x} \in \mathbf{X} \mid \forall \mathbf{p} \in \mathbf{P}, \mathbf{g}(\mathbf{x}, \mathbf{p}) \leq \mathbf{0}\}, \quad (3.10)$$

resolver (3.9) consiste em encontrar o conjunto minimizadores robustos  $\mathbf{X}^* \subseteq \mathbf{X}'$  para o pior caso da ação das incertezas. Formalmente,

$$\mathbf{X}^* = \{\mathbf{x}^* \in \mathbf{X}' \mid \nexists \mathbf{x} \in \mathbf{X}', \max_{\mathbf{p} \in \mathbf{P}} \mathbf{f}(\mathbf{x}, \mathbf{p}) \preceq \max_{\mathbf{p}' \in \mathbf{P}} \mathbf{f}(\mathbf{x}^*, \mathbf{p}')\}. \quad (3.11)$$

A próxima etapa é apresentar as imagens do espaço de busca sujeito à ação das incertezas. Considere as imagens das funções  $\mathbf{y} = \mathbf{f}(\mathbf{x}, \mathbf{p})$  e  $\mathbf{Y}' = \mathbf{f}(\mathbf{X}', \mathbf{P})$ , sendo

$$\mathbf{f}(\mathbf{X}', \mathbf{P}) = \bigcup_{\mathbf{x} \in \mathbf{X}', \mathbf{p} \in \mathbf{P}} \mathbf{f}(\mathbf{x}, \mathbf{p}). \quad (3.12)$$

No entanto, como consequência do Teo.1, se para alguma solução  $\mathbf{x}^*$  o processo de maximização em (3.9) resultar em  $\mathbf{u}^{\max}(\mathbf{f}(\mathbf{x}^*, \mathbf{p})) \notin \Upsilon_{\succeq}(\mathbf{f}(\mathbf{x}^*, \mathbf{p}))$ , então pode ocorrer que  $\mathbf{u}^{\max}(\mathbf{f}(\mathbf{X}^*, \mathbf{P})) \notin \mathbf{Y}'$ . Portanto, para representar a imagem robusta de todas as soluções  $\mathbf{X}^*$  do problema (3.9) é necessário

definir um espaço que contenha  $\mathbf{u}^{\max}(\mathbf{f}(\mathbf{x}^*, \mathbf{p}))$ ,  $\forall \mathbf{x}^* \in \mathbf{X}^*$  e  $\forall \mathbf{p} \in \mathbf{P}$ . Isso é feito substituindo-se a imagem  $\mathbf{f}(\mathbf{X}', \mathbf{P})$  pela caixa  $[\mathbf{Y}]'$  definida como

$$\overline{[Y_i]}' = \max_{\mathbf{p} \in \mathbf{P}} f_i(\mathbf{x}, \mathbf{p}), \quad \forall \mathbf{x} \in \mathbf{X}', \quad i = \{1, \dots, n_f\}; \quad (3.13)$$

$$\underline{[Y_i]}' = \min_{\mathbf{p} \in \mathbf{P}} f_i(\mathbf{x}, \mathbf{p}), \quad \forall \mathbf{x} \in \mathbf{X}', \quad i = \{1, \dots, n_f\}. \quad (3.14)$$

Agora, seja  $\mathbf{Y}^*$  a *fronteira Pareto robusta*, e  $\mathbf{Y}^+$  e  $\mathbf{Y}^-$  as *porções acima e abaixo da fronteira robusta*, respectivamente. As imagens  $\mathbf{Y}^*$ ,  $\mathbf{Y}^+$  e  $\mathbf{Y}^-$  associam-se da forma

$$[\mathbf{Y}]' = \mathbf{Y}^- \cup \mathbf{Y}^+, \quad \mathbf{Y}^* \subseteq \mathbf{Y}^+, \quad (3.15)$$

sendo:

$$\mathbf{Y}^+ = \{\mathbf{y}^+ \in [\mathbf{Y}]' \mid \exists \mathbf{x} \in \mathbf{X}, \forall \mathbf{p} \in \mathbf{P}, \mathbf{f}(\mathbf{x}, \mathbf{p}) \preceq \mathbf{y}^+\}; \quad (3.16)$$

$$\mathbf{Y}^- = \{\mathbf{y}^- \in [\mathbf{Y}]' \mid \mathbf{y} \in [\mathbf{Y}]', \mathbf{y} \notin \mathbf{Y}^+\}; \quad (3.17)$$

$$\mathbf{Y}^* = \{\mathbf{y}^* \in [\mathbf{Y}]' \mid \nexists \mathbf{y} \in \mathbf{Y}^+, \mathbf{y} \preceq \mathbf{y}^*\}. \quad (3.18)$$

Na Fig. 3.1, o lado direito ilustra as regiões de  $\mathbf{Y}^*$ ,  $\mathbf{Y}^+$  e  $\mathbf{Y}^-$ ; o lado esquerdo, apresenta o espaço de busca e o espaço das incertezas. A título de exemplo, o par  $(\mathbf{x}_0^*, \mathbf{p}_0^*)$  estabelece relação entre os espaços de busca e de incertezas com o espaço dos objetivos. Na seqüência, são apresentados dois exemplos.

**Exemplo 12** *Problema com soluções robustas pertencentes à imagem  $\mathbf{Y}'$ .*

$$\begin{aligned} \min_{\mathbf{x} \in [-1, 1]} \max_{\mathbf{p} \in [0, 0.25]} \quad & f_1(\mathbf{x}, \mathbf{p}) = x_1^2 + x_2^2 + p_1, \\ & f_2(\mathbf{x}, \mathbf{p}) = (x_1 - 1)^2 + x_2^2 + p_2. \end{aligned} \quad (3.19)$$

*Problema irrestrito, logo  $\mathbf{X} = \mathbf{X}'$ . O espaço dos objetivos está plotado na Fig. 3.2(a). Os minimizadores robustos estão localizados em  $x_1 \in [0, 1]$  e  $x_2 = 0$ . O pior caso acontece quando  $\mathbf{p} = \mathbf{1}$ . Este exemplo apresenta as regiões robustas claramente separadas.*

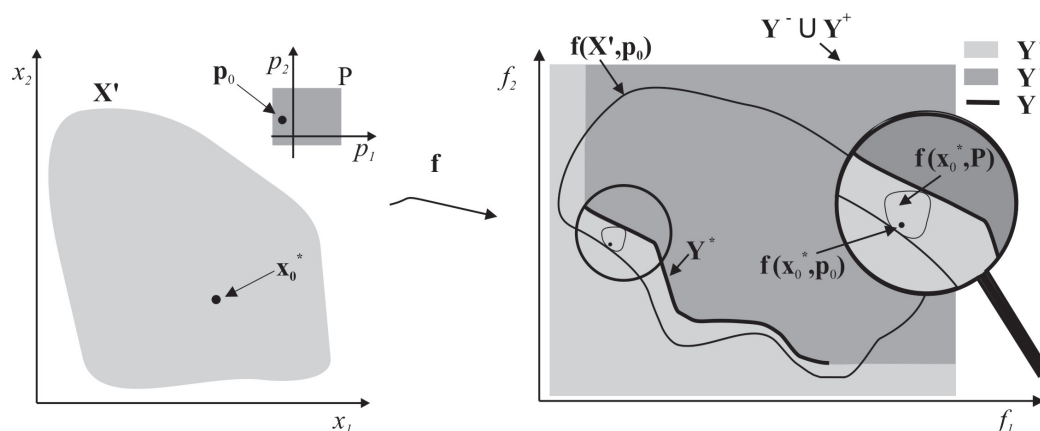
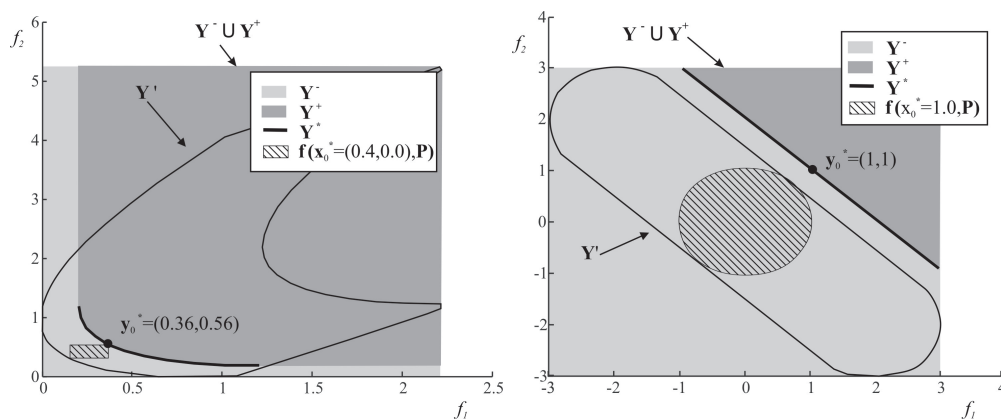


Fig. 3.1: Os espaços de busca, de incerteza e dos objetivos no problema de otimização multi-objetivo robusto.



(a) Problema onde  $Y^* \subset Y^-$ .

(b) Problema onde  $Y^* \not\subset Y^-$ .

Fig. 3.2: Exemplos de problema multi-objetivo robusto. Observe em (b) que toda a fronteira robusta está fora da imagem das funções objetivo.

**Exemplo 13** *Problema com soluções robustas não pertencentes à imagem  $Y^-$ .*

$$\begin{aligned} \min_{x \in [-2, 2]} \max_{\|\mathbf{p}\| \leq 1} \quad & f_1(x, p_1) = -x + p_1, \\ & f_2(x, p_2) = x + p_2. \end{aligned} \tag{3.20}$$

Problema irrestrito, logo  $\mathbf{X} = \mathbf{X}'$ . O pior caso acontece quando  $\|\mathbf{p}\| = 1$  para qualquer instante da variável  $x$ . Como mostrado na Fig. 3.2(b), a fronteira robusta  $Y^* \not\subset Y^-$ .

### 3.4 Algoritmo Intervalar Multi-Objetivo Robusto I

#### 3.4.1 Preliminares

Encontrar a fronteira de Pareto robusta  $\mathbf{Y}^*$  e o conjunto de minimizadores  $\mathbf{X}^*$  pode ser uma tarefa dispendiosa em termos de esforço computacional e, difícil de ser concluída com êxito. Entretanto, computar envelopes precisos que contenham  $\mathbf{Y}^*$  e  $\mathbf{f}(\mathbf{X}^*, \mathbf{P})$  é mais fácil e considerou-se uma contribuição importante. O primeiro método de otimização desta Tese computa estes envelopes, e sua precisão depende diretamente da representação de  $\mathbf{Y}^*$  que é descrita nas próximas definições.

**Definição 19** Considere  $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$ ,  $\mathbf{U} \subset \mathbf{Y}^-$ . Então, a representação computável de  $\mathbf{Y}^-$  associada com  $\mathbf{U}$  é

$$\mathbf{K}^- = \{\mathbf{u}' \in [\mathbf{Y}]' \mid \exists \mathbf{u} \in \Upsilon_{\succeq}(\mathbf{U}), \mathbf{u}' \preceq \mathbf{u}\}. \quad (3.21)$$

Analogamente, se  $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m\}$ ,  $\mathbf{V} \subset \mathbf{Y}^+$ , então a representação computável de  $\mathbf{Y}^+$  associada com  $\mathbf{V}$  é

$$\mathbf{K}^+ = \{\mathbf{v}' \in [\mathbf{Y}]' \mid \exists \mathbf{v} \in \Upsilon_{\preceq}(\mathbf{V}), \mathbf{v}' \succeq \mathbf{v}\}. \quad (3.22)$$

**Definição 20** A fronteira superior de  $\mathbf{K}^-$  e a fronteira inferior de  $\mathbf{K}^+$  são definidas como

$$\partial\mathbf{K}^- = \{\mathbf{u} \in \mathbf{K}^- \mid \nexists \mathbf{u}' \in \mathbf{K}^-, \mathbf{u} \preceq \mathbf{u}'\}; \quad (3.23)$$

$$\partial\mathbf{K}^+ = \{\mathbf{v} \in \mathbf{K}^+ \mid \nexists \mathbf{v}' \in \mathbf{K}^+, \mathbf{v}' \preceq \mathbf{v}\}. \quad (3.24)$$

**Definição 21** Simbolizando  $\setminus$  como operador complemento, as porções não computáveis diretamente de  $\mathbf{Y}^-$  e  $\mathbf{Y}^+$  são definidas por

$$\Delta\mathbf{K}^- = (\mathbf{Y}^- \setminus \mathbf{K}^-) \cup \partial\mathbf{K}^-; \quad (3.25)$$

$$\Delta\mathbf{K}^+ = (\mathbf{Y}^+ \setminus \mathbf{K}^+) \cup \partial\mathbf{K}^+. \quad (3.26)$$

**Teorema 2** [Envelope sobre a fronteira Pareto robusta] Dada a precisão para computação de  $\mathbf{K}^-$  e  $\mathbf{K}^+$ , a seguinte inclusão é sempre verdadeira

$$\mathbf{Y}^* \subset \Delta\mathbf{K}^- \cup \Delta\mathbf{K}^+. \quad \blacksquare$$

**Prova:** De (3.15),  $[\mathbf{Y}]' = \mathbf{Y}^- \cup \mathbf{Y}^+$ . De (3.25) e (3.26),  $\mathbf{Y}^- = \mathbf{K}^- \cup \Delta\mathbf{K}^-$ ,  $\mathbf{K}^- \cap \Delta\mathbf{K}^- = \partial\mathbf{K}^-$  e  $\mathbf{Y}^+ = \mathbf{K}^+ \cup \Delta\mathbf{K}^+$ ,  $\mathbf{K}^+ \cap \Delta\mathbf{K}^+ = \partial\mathbf{K}^+$ . Substituindo  $\mathbf{Y}^-$  e  $\mathbf{Y}^+$  em (3.15), resulta em

$$[\mathbf{Y}]' = (\mathbf{K}^- \cup \Delta\mathbf{K}^-) \cup (\mathbf{K}^+ \cup \Delta\mathbf{K}^+).$$

Mas,  $\mathbf{K}^- \cap \Delta\mathbf{K}^- = \partial\mathbf{K}^-$  e  $\mathbf{K}^+ \cap \Delta\mathbf{K}^+ = \partial\mathbf{K}^+$  e portanto, há ocorrências extras de  $\partial\mathbf{K}^-$  e  $\partial\mathbf{K}^+$  que podem ser removidas. Então,

$$\begin{aligned} [\mathbf{Y}]' &= ((\mathbf{K}^- \setminus \partial\mathbf{K}^-) \cup \Delta\mathbf{K}^-) \cup ((\mathbf{K}^+ \setminus \partial\mathbf{K}^+) \cup \Delta\mathbf{K}^+); \\ [\mathbf{Y}]' &= ((\mathbf{K}^- \setminus \partial\mathbf{K}^-) \cup (\mathbf{K}^+ \setminus \partial\mathbf{K}^+)) \cup (\Delta\mathbf{K}^- \cup \Delta\mathbf{K}^+). \end{aligned}$$

As expressões (3.18), (3.21) e (3.22) asseguram que  $\mathbf{Y}^* \not\subset (\mathbf{K}^- \setminus \partial\mathbf{K}^-)$  e  $\mathbf{Y}^* \not\subset (\mathbf{K}^+ \setminus \partial\mathbf{K}^+)$ . Então,

$$\mathbf{Y}^* \subset \Delta\mathbf{K}, \text{ sendo } \Delta\mathbf{K} = \Delta\mathbf{K}^- \cup \Delta\mathbf{K}^+. \quad \blacksquare$$

De fato, (3.18) garante que  $\mathbf{Y}^* \subset \Delta\mathbf{K}^+$ . Mas  $\Delta\mathbf{K}^+$  sozinho não pode ser diretamente envelopado. A Fig. 3.3 ilustra a representação das aproximações das regiões robustas.

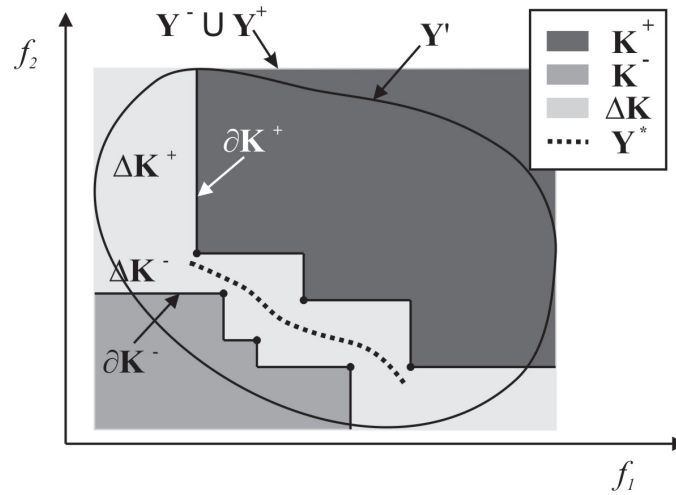


Fig. 3.3: Representação computável das regiões robustas.

**Teorema 3 (Envelope sobre os minimizadores robustos)** *A seguinte inclusão é sempre verdadeira*

$$\mathbf{f}(\mathbf{X}^*, \mathbf{P}) \subset \mathbf{K}^- \cup \Delta\mathbf{K}^- \cup \Delta\mathbf{K}^+. \quad \blacksquare$$

**Prova:** De (3.15),  $[\mathbf{Y}]' = \mathbf{Y}^- \cup \mathbf{Y}^+$ . De (3.16),  $\mathbf{f}(\mathbf{X}^*, \mathbf{P}) \not\subset \mathbf{Y}^+$ . Desse modo,  $\mathbf{f}(\mathbf{X}^*, \mathbf{P}) \subset [\mathbf{Y}]' \setminus \mathbf{Y}^+$ . Como  $\mathbf{Y}^+ = \mathbf{K}^+ \cup \Delta\mathbf{K}^+$  e somente  $\mathbf{K}^+$  pode ser diretamente computável, então é também verdade

$$\begin{aligned} \mathbf{f}(\mathbf{X}^*, \mathbf{P}) &\subset [\mathbf{Y}]' \setminus \mathbf{K}^+; \\ \mathbf{f}(\mathbf{X}^*, \mathbf{P}) &\subset \mathbf{K}^- \cup \Delta\mathbf{K}^- \cup \Delta\mathbf{K}^+; \\ \mathbf{f}(\mathbf{X}^*, \mathbf{P}) &\subset \mathbf{K}^- \cup \Delta\mathbf{K}, \text{ sendo } \Delta\mathbf{K} = \Delta\mathbf{K}^- \cup \Delta\mathbf{K}^+. \blacksquare \end{aligned}$$

Estes dois teoremas suportam o algoritmo apresentado na próxima subseção a envelopar a fronteira de Pareto robusta, segundo dada precisão. A validação é feita no capítulo seguinte.

### 3.4.2 [I]RMOA I

O primeiro algoritmo para otimização robusta proposto aqui, [I]RMOA I (Interval Robust Multi-Objective Algorithm) [112], é uma técnica heurística baseada inteiramente em Análise Intervalar desenvolvida para envelopar  $\mathbf{Y}^*$ . Em poucas palavras, o algoritmo inicia com um conjunto de amostras uniformemente distribuídas no espaço de busca e aleatoriamente no espaço das incertezas. As soluções são avaliadas pela função  $\mathbf{f}(\mathbf{x}, \mathbf{p})$  e seus resultados são armazenados em uma lista encadeada tipo fila. Cada elemento desta fila é testado se ele pertence à  $\mathbf{K}^-$  ou  $\mathbf{K}^+$ . Após a classificação completa, [I]RMOA I tem informação suficiente para encontrar  $\Delta\mathbf{K}$ . Por exemplo,  $\Delta\mathbf{K} = [\mathbf{Y}]' \setminus (\mathbf{K}^+ \cup \mathbf{K}^-)$ , e conseqüentemente  $\Delta\mathbf{K} \supset \mathbf{Y}^*$ . Em adição, a região definida por  $\mathbf{K}^-$  e  $\Delta\mathbf{K}$  contém  $\mathbf{f}(\mathbf{X}^*, \mathbf{P})$ .

Considerando a existência de suporte computacional para a aritmética dos métodos intervalares apresentados na Seção 2.3, a implementação do [I]RMOA I foi dividida em duas partes: a) o algoritmo principal e; b) o algoritmo para as funções FPS (feasible point searcher) e CSC (computable sufficient conditions). FPS e CSC foram introduzidas por Jaulin *et al.* em [64] para dividir intervalos e resolver aplicações na área de controle. Aqui, estas funções são usadas no contexto de otimização. O pseudo código do [I]RMOA I está na Tab.3.1 e seus símbolos na Tab.3.2. Neste método, a estrutura de dados principal são filas. A precisão final do método é assegurada por dois parâmetros  $\varepsilon$  declarados na entrada. O passo 1 contém as inicializações das filas. No passo 2, o espaço de busca é amostrado pela função *grade*, que divide cada dimensão do espaço de busca em fatias de tamanho máximo  $\varepsilon_{\mathbf{x}\#}$ . Para cada amostra  $\mathbf{x}$ , um valor de incerteza  $\mathbf{p}$  é escolhido aleatoriamente. Todos os pares  $(\mathbf{x}, \mathbf{p})$  da função *grade* são avaliados pelas funções objetivo e inseridos em  $\mathbf{Q}_{\mathbf{fxp}}$ . O laço principal do algoritmo, entre os passos 3 e 14, se mantém ativo enquanto  $\mathbf{Q}_{\mathbf{fxp}}$  for não vazio. Nos passos 4, 5 e 6

Tab. 3.1: Algoritmo [I]RMOA I.

	<b>entrada</b> ( $[\mathbf{X}], [\mathbf{P}], \varepsilon_{\mathbf{x}\#}, \varepsilon_{[\mathbf{x}]}$ )
	<b>saída</b> ( $\mathbf{K}^-, \mathbf{K}^+$ )
1	inicialize $\mathbf{Q}_{\mathbf{fxp}}, \mathbf{Q}_{\mathbf{K}^+}, \mathbf{Q}_{\mathbf{K}^-}$ com $\emptyset$ e $\mathbf{Q}_{[\mathbf{x}]}$ com $[\mathbf{X}]$
2	grade( $[\mathbf{X}], \varepsilon_{\mathbf{x}\#}, [\mathbf{P}]$ ), avalie as amostras usando $\mathbf{f}(\mathbf{x}, \mathbf{p})$ e armazene os resultados em $\mathbf{Q}_{\mathbf{fxp}}$
3	se $\mathbf{Q}_{\mathbf{fxp}} = \emptyset$ então siga para 15
4	retire $\mathbf{f}(\mathbf{x}, \mathbf{p})$ de $\mathbf{Q}_{\mathbf{fxp}}$
5	se existe um $\mathbf{v}$ em $\mathbf{Q}_{\mathbf{K}^+}$ tal que $\mathbf{v} \preceq \mathbf{f}(\mathbf{x}, \mathbf{p})$ então siga para 3
6	se existe um $\mathbf{u}$ em $\mathbf{Q}_{\mathbf{K}^-}$ tal que $\mathbf{f}(\mathbf{x}, \mathbf{p}) \preceq \mathbf{u}$ então siga para 3
7	chame FPS( $[\mathbf{P}], [\mathbf{X}], \mathbf{f}(\mathbf{x}, \mathbf{p}), \varepsilon_{[\mathbf{x}]}$ )
8	se FPS retornar $\mathbf{x}^*$
9	descarte todos $\mathbf{v}$ em $\mathbf{Q}_{\mathbf{K}^+}$ tais que $\mathbf{f}(\mathbf{x}, \mathbf{p}) \preceq \mathbf{v}$
10	insira $\mathbf{f}(\mathbf{x}, \mathbf{p})$ em $\mathbf{Q}_{\mathbf{K}^+}$
11	senão
12	descarte todos $\mathbf{u}$ em $\mathbf{Q}_{\mathbf{K}^-}$ tal que $\mathbf{u} \preceq \mathbf{f}(\mathbf{x}, \mathbf{p})$
13	insira $\mathbf{f}(\mathbf{x}, \mathbf{p})$ em $\mathbf{Q}_{\mathbf{K}^-}$
14	siga para 3
15	compute $\mathbf{K}^-$ usando os elementos de $\mathbf{Q}_{\mathbf{K}^-}$ em (3.21)
16	compute $\mathbf{K}^+$ usando os elementos de $\mathbf{Q}_{\mathbf{K}^+}$ em (3.22)
17	retorne $\mathbf{K}^-, \mathbf{K}^+$

os pontos são avaliados e comparados com todos os elementos em  $\mathbf{Q}_{\mathbf{K}^+}$  e/ou  $\mathbf{Q}_{\mathbf{K}^-}$ . Se  $\mathbf{f}(\mathbf{x}, \mathbf{p})$  é maior que algum ponto em  $\mathbf{Q}_{\mathbf{K}^+}$ , ele certamente pertence a  $\mathbf{K}^+$  e pode ser descartado. Também, se  $\mathbf{f}(\mathbf{x}, \mathbf{p})$  é menor que algum ponto em  $\mathbf{Q}_{\mathbf{K}^-}$  ele certamente pertence a  $\mathbf{K}^-$  e pode ser descartado. No início, as filas  $\mathbf{Q}_{\mathbf{K}^+}$  e  $\mathbf{Q}_{\mathbf{K}^-}$  são vazias e o algoritmo é dirigido a 7. Os procedimentos 5 e 6 evitam esforço computacional extra no passo 7, A função FPS e sua sub-função CSC dividem o espaço de busca armazenado em  $\mathbf{Q}_{[\mathbf{x}]}$  e a incerteza  $[\mathbf{P}]$  para encontrar um minimizador para  $\mathbf{f}(\mathbf{x}, \mathbf{p})$ . Se o teste retorna “ $\mathbf{x}^*$ ”, então  $\mathbf{f}(\mathbf{x}, \mathbf{p})$  tem que ser inserido em  $\mathbf{Q}_{\mathbf{K}^+}$  (passo 10), por outro lado, FPS retorna “sem solução” e  $\mathbf{f}(\mathbf{x}, \mathbf{p})$  tem que ser inserido em  $\mathbf{Q}_{\mathbf{K}^-}$  (passo 13). Antes da inserção, novas comparações são executadas entre  $\mathbf{f}(\mathbf{x}, \mathbf{p})$  com sua respectiva fila para eliminar pontos redundantes (passos 9 e 12). O passo 14 marca o final da inserção de uma iteração e conduz o algoritmo para o passo 3. Depois da classificação de todo elemento em  $\mathbf{Q}_{\mathbf{x}}$ , [I]RMOA I sai do laço principal retorna os pontos que definem  $\mathbf{K}^-$  e  $\mathbf{K}^+$  (passos 15 e 16). As versões origi-

Tab. 3.2: Símbolos nos algoritmos [I]RMOA I, FPS e CSC.

Símbolos	Descrição / Conteúdo	Tipo
$[\mathbf{X}]$	espaço de busca inicial	$\mathbb{I}\mathbb{R}^{n_x}$
$[\mathbf{P}]$	espaço de incerteza inicial	$\mathbb{I}\mathbb{R}^{n_p}$
$[\mathbf{p}]$	parâmetro de incerteza	$\mathbb{I}\mathbb{R}^{n_p}$
$[\mathbf{x}]$	parâmetro de busca	$\mathbb{I}\mathbb{R}^{n_x}$
$\mathbf{x}^*$	minimizador para $\mathbf{f}(\mathbf{x}, \mathbf{p})$	$\mathbb{R}^{n_x}$
$\mathbf{f}$	função objetivo	$\mathbb{R}^{n_f}$
$\mathbf{y}$	resultado de $\mathbf{f}(\mathbf{x}, \mathbf{p})$	$\mathbb{R}^{n_f}$
$[\mathbf{f}]$	função de inclusão para $\mathbf{f}$	$\mathbb{I}\mathbb{R}^{n_f}$
$\varepsilon_{\mathbf{x}\#}$	mínima precisão para a função grade	$\mathbb{R}$
$\varepsilon_{[\mathbf{x}]}$	mínima largura para $[\mathbf{x}]$	$\mathbb{R}$
$\mathbf{Q}_{\mathbf{fxp}}$	amostras avaliadas da função grade	Fila: $\mathbb{R}^{n_f}$
$\mathbf{Q}_{[\mathbf{x}]}$	subpavimentos do parâmetro de busca	Fila: $\mathbb{I}\mathbb{R}^{n_x}$
$\mathbf{Q}$	cópia de $\mathbf{Q}_{[\mathbf{x}]}$	Fila: $\mathbb{I}\mathbb{R}^{n_x}$
$\mathbf{Q}_{\mathbf{K}^+}$	pontos que definem $\mathbf{K}^+$	Fila: $\mathbb{R}^{n_f}$
$\mathbf{Q}_{\mathbf{K}^-}$	pontos que definem $\mathbf{K}^-$	Fila: $\mathbb{R}^{n_f}$
$\mathbf{S}_{[\mathbf{p}]}$	subpavimentos do parâmetro de incerteza	Pilha: $\mathbb{I}\mathbb{R}^{n_p}$

inais do FPS (Tab.3.3) e do CSC (Tab.3.4) em [64] foram adaptadas para [I]RMOA I. Detalhes técnicos e provas de convergência são encontradas na referência citada. Por agora, FPS bissecta o espaço de busca e tenta encon-

Tab. 3.3: Algoritmo FPS.

<b>entrada</b>	$([\mathbf{P}], [\mathbf{X}], \mathbf{y}, \varepsilon_{[\mathbf{x}]})$
<b>saída</b>	$(\mathbf{x}^*$ ou “sem solução”)
1	inicialize $\mathbf{Q}$ com $[\mathbf{X}]$
2	retire $[\mathbf{x}]$ de $\mathbf{Q}_{[\mathbf{x}]}$
3	se $w([\mathbf{x}]) \leq \varepsilon_{[\mathbf{x}]}$ siga para 9
4	chame $\text{CSC}([\mathbf{x}], \mathbf{y}, [\mathbf{P}], \varepsilon_{[\mathbf{x}]})$
5	se CSC retornar $\mathbf{x}^*$ então insira $[\mathbf{x}]$ em $\mathbf{Q}_{[\mathbf{x}]}$ e retorne $\mathbf{x}^*$ . Fim.
6	se CSC retornar “sem solução” então siga para 8
7	bissecte $[\mathbf{x}]$ e insira as duas caixas em $\mathbf{Q}_{[\mathbf{x}]}$
8	se $\mathbf{Q}_{[\mathbf{x}]}$ é não vazio então siga para 2
9	retorne “sem solução”. Fim.

trar um minimizador para  $\mathbf{f}(\mathbf{x}, \mathbf{p})$ , considerando as incertezas. Então, FPS começa com a primeira caixa  $[\mathbf{x}]$  de  $\mathbf{Q}_{[\mathbf{x}]}$  e  $[\mathbf{P}]$  e chama CSC exaustivamente até CSC retornar um ponto solução “ $\mathbf{x}^*$ ” ou “sem solução”. Se CSC retornar “sem conclusão” então FPS bissecta  $[\mathbf{x}]$  e o processo é repetido. A precisão deste procedimento é definida pelo usuário na escolha de  $\varepsilon_{[\mathbf{x}]}$ . Na subrotina CSC, um processo similar a FPS acontece, mas somente o parâmetro de incerteza  $[\mathbf{P}]$  é bissectado. Quando CSC retorna um ponto solução “ $\mathbf{x}^*$ ” ou “sem solução” para FPS, FPS reenvia a informação para [I]RMOA I e o laço principal recomeça.

Tab. 3.4: Algoritmo CSC.

<b>entrada</b>	$([\mathbf{x}], \mathbf{y}, [\mathbf{P}], \varepsilon_{[\mathbf{x}]})$
<b>saída</b>	$(\mathbf{x}^*$ ou “sem conclusão” ou “sem solução”)
1	empilhe $[\mathbf{P}]$ em $\mathbf{S}_{[\mathbf{p}]}$ , $prossiga \leftarrow verdadeiro$
2	desempilhe de $\mathbf{S}_{[\mathbf{p}]}$ em $[\mathbf{p}]$
3	se $[\mathbf{f}]([\mathbf{x}], \mathbf{c}([\mathbf{p}])) \succeq \mathbf{y}$ então retorne “sem solução”. Fim.
4	se $[\mathbf{f}](\mathbf{c}([\mathbf{x}]), [\mathbf{p}]) \preceq \mathbf{y}$ então siga para 7
5	se $w([\mathbf{p}]) < \varepsilon_{[\mathbf{x}]}$ então $prossiga \leftarrow falso$ e siga para 7
6	bissecte $[\mathbf{p}]$ e empilhe em $\mathbf{S}_{[\mathbf{p}]}$ as duas caixas resultantes
7	se $\mathbf{S}_{[\mathbf{p}]}$ é não vazio então vai para 2
8	se $prossiga = verdadeiro$ então $\mathbf{x}^* \leftarrow \mathbf{c}([\mathbf{x}])$ e retorne $\mathbf{x}^*$ . Fim.
9	retorne “sem conclusão”. Fim.

A qualidade das funções de inclusão e o parâmetro de precisão  $\varepsilon_{[\mathbf{x}]}$  são os responsáveis pelo resultado correto dos algoritmos FPS e CSC, e conseqüentemente, do [I]RMOA I. Usualmente, se as relações  $\varepsilon_{[\mathbf{x}]} < w([\mathbf{P}])$  e  $\varepsilon_{[\mathbf{x}]} < \frac{\varepsilon_{\#}}{2}$  puderem ser asseguradas o resultado de FPS e CSC é satisfatório. A primeira condição garante bissecções no espaço de incertezas e a segunda permite caixas menores que a precisão da amostragem inicial sejam analisadas.

Se a memória de máquina for suficiente para armazenar as estruturas de dados, garante-se que [I]RMOA I converge (Tab.3.1) porque todos os processos são finitos. A função grade no passo 2 é um processo finito baseado na divisão do espaço de busca. O laço principal no passo 3 e seus laços internos 5, 6, 9 e 12 são baseados em filas com finitos elementos. Finalmente, os detalhes de convergência no passo 7 são discutidos pelos autores de FPS em [64].

### 3.5 Algoritmo Intervalar Multi-Objetivo Robusto II

A expressão formal que define o problema robusto (3.9) deve ser interpretada como a maximização da interferência das incertezas sobre as funções objetivo, seguida pela minimização destes resultados. Neste sentido, o problema multi-objetivo robusto se assemelha a resolução de dois problemas multi-objetivos. Mas existe uma particularidade a comentar: para cada ponto viável  $\mathbf{x}$ , usualmente a fase de maximização resulta em conjunto de pontos, sendo o ponto ideal para maximização deste conjunto considerado como a imagem robusta para  $\mathbf{x}$ . Assim, considerando a computação de incertezas sobre todo o espaço de busca viável, o conjunto Pareto robusto é formado pelo menores pontos ideais para maximização. O [I]RMOA II (Interval Robust Multi-Objective Algorithm II) computa os pontos ideais explorando conceitos da Análise Intervalar. O cálculo dos pontos ideais e outros detalhes que suportam o [I]RMOA II são explicados nas próximas subseções.

#### 3.5.1 Preliminares

**Processo de maximização e ponto ideal:** O algoritmo [I]RMOA II maximiza a ação das incertezas sobre os pontos viáveis com a finalidade de descobrir o ponto ideal de maximização. O processo para computar o ponto ideal inicia com a divisão do domínio das incertezas  $\mathbf{P}$ , convenientemente descrito na forma  $[\mathbf{P}]$ , em  $n_{sp}$  subpavimentos sem sobreposição  $[\mathbf{p}]_i$ ,

$$[\mathbf{P}] = \bigcup_{i=1, \dots, n_{sp}} [\mathbf{p}]_i. \quad (3.27)$$

O algoritmo da Tab.3.5 apresenta como foi implementado a divisão do domínio de incertezas  $[\mathbf{P}]$  em subpavimentos com precisão  $\varepsilon_{[\mathbf{p}]}$ . Os subpavimentos gerados foram armazenados na fila  $\mathbf{Q}_{[\mathbf{p}]}$ .

Tab. 3.5: Divisão de domínio em subpavimentos.

	<b>entrada</b> ( $[\mathbf{P}], \varepsilon_{[\mathbf{p}]}$ )
	<b>saída</b> ( $\mathbf{Q}_{[\mathbf{p}]}$ )
1	inicialize $\mathbf{Q}_{[\mathbf{p}]}$ com $[\mathbf{P}]$
2	se $w(\mathbf{Q}_{[\mathbf{p}]\{1\}}) \leq \varepsilon_{[\mathbf{p}]}$ , siga para 5
3	retire $[\mathbf{p}]$ de $\mathbf{Q}_{[\mathbf{p}]}$
4	bissecte $[\mathbf{p}]$ , insira as duas caixas resultantes em $\mathbf{Q}_{[\mathbf{p}]}$ e siga para 2
5	retorne $\mathbf{Q}_{[\mathbf{p}]}$

Assim, considerando a representação de  $[\mathbf{P}]$  por subpavimentos  $[\mathbf{p}]$  armazenados em  $\mathbf{Q}_{[\mathbf{p}]}$ , considerando que  $\mathbf{x}$  seja uma variável de busca viável e que  $[\mathbf{f}](\mathbf{x}, [\mathbf{P}])$  represente sua função de inclusão, a rotina MIF( $\mathbf{x}, \mathbf{Q}_{[\mathbf{p}]}$ ) (Minimal Inclusion Function), que representa  $[\mathbf{f}]_{\mathbf{x}}^*$ , é computada como

$$\overline{[f_i]_{\mathbf{x}}^*} = \max_{[\mathbf{p}] \in \mathbf{Q}_{[\mathbf{p}]}} \overline{[f_i]}(\mathbf{x}, [\mathbf{p}]), \quad i = \{1, \dots, n_f\}; \quad (3.28)$$

$$\underline{[f_i]_{\mathbf{x}}^*} = \min_{[\mathbf{p}] \in \mathbf{Q}_{[\mathbf{p}]}} \underline{[f_i]}(\mathbf{x}, [\mathbf{p}]), \quad i = \{1, \dots, n_f\}. \quad (3.29)$$

Analogamente, a rotina MIF( $[\mathbf{x}], \mathbf{Q}_{[\mathbf{p}]}$ ) computa a função de inclusão mínima para o intervalo  $[\mathbf{x}]$ . O ponto ideal para maximização, expressado por  $\overline{[\mathbf{f}]_{\mathbf{x}}^*}$ , é a imagem robusta de  $\mathbf{x}$ . Similarmente,  $\overline{[\mathbf{f}]_{[\mathbf{x}]}}^*$  é a imagem robusta do intervalo  $[\mathbf{x}]$ . A Fig. 3.4 ilustra a computação do ponto ideal para maximização de um intervalo  $[\mathbf{x}] \in \mathbb{I}\mathbb{R}^{n_x}$  e de um de seus pontos internos  $\mathbf{x}_0 \in [\mathbf{x}]$ . Em alguns casos, a função de inclusão mínima pode ser computada de uma única vez. De acordo com a Subseção 2.3.8, se cada variável ocorre no máximo uma vez na formulação que vai ser computada, então a função natural é mínima.

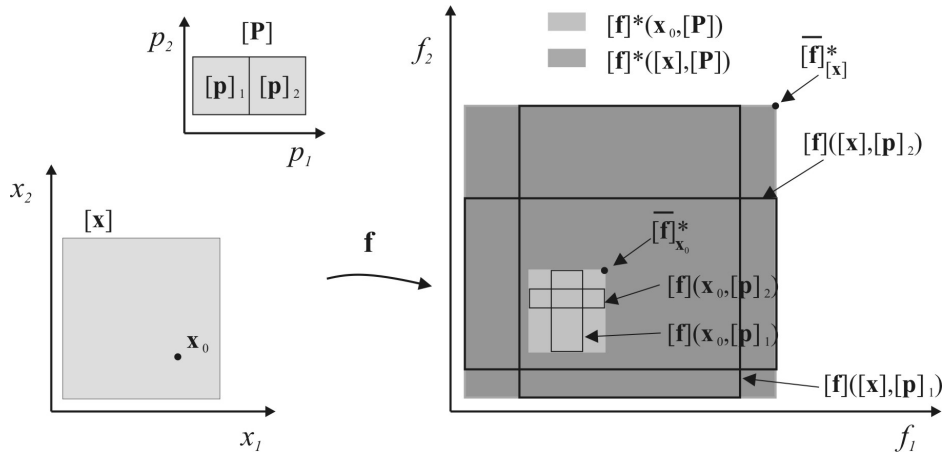


Fig. 3.4: Computação dos pontos ideais de maximização para intervalos  $[\mathbf{f}]_{[\mathbf{x}]}^*$  e para pontos  $[\mathbf{f}]_{\mathbf{x}}^*$  quando sujeitos à incerteza  $[\mathbf{P}]$ .

**Tratamento de restrições:** O desenvolvimento desse tópico considera que as funções de restrição são escritas no formato  $\mathbf{g} \leq 0$ . Sendo utilizadas, para intervalos, a notação  $[\mathbf{g}] \leq 0$  ou equivalentemente  $[\mathbf{g}] \subseteq [-\infty, 0]$ . O [I]RMOA II exclui espaço de busca não viável empregando testes de inclusão [t] sobre o vetor de funções de restrição  $[\mathbf{g}]$ . Assim, considerando os parâmetros de busca e de incerteza no formato de intervalos, o teste de

inclusão é expressado como

$$[t]([\mathbf{x}]) = \begin{cases} 1, & \text{caso } [\mathbf{g}]([\mathbf{x}], [\mathbf{p}]) \subseteq [-\infty, 0], \\ 0, & \text{caso } [\mathbf{g}]([\mathbf{x}], [\mathbf{p}]) \subset [0, \infty], \text{ e} \\ [0, 1], & \text{por outro lado.} \end{cases} \quad (3.30)$$

Assim,  $\forall \mathbf{x} \in [\mathbf{x}]$ ,  $\mathbf{x}$  é viável se  $[t]([\mathbf{x}]) = 1$  e não viável se  $[t]([\mathbf{x}]) = 0$ . Por outro lado, nada se pode concluir sobre os elementos de  $[\mathbf{x}]$ , e ele deverá ser encaminhado a bissecção. As caixas resultantes da bissecção são também submetidas ao teste  $[t]([\mathbf{x}])$ . Um parâmetro de esforço  $\varepsilon_{[\mathbf{x}]}$ , por exemplo  $w([\mathbf{x}]) < \varepsilon_{[\mathbf{x}]}$ , marca o fim das bissecções e o tomador de decisão deve decidir sobre o futuro de todas as caixas  $[\mathbf{x}]$  tais que  $w([\mathbf{x}]) < \varepsilon_{[\mathbf{x}]}$ . Para o [I]RMOA II, optou-se pela exclusão de todas as soluções cuja condição acima não tenha sido satisfeita. Observe que o procedimento acima foi baseado no método SIVIA (veja 2.3.12). O algoritmo para tratamento de restrições está apresentado na Tab.3.6 e o Exem.14 mostra um caso geral.

Tab. 3.6: Algoritmo para tratamento de restrições.

<b>entrada</b> ( $[\mathbf{X}], [\mathbf{P}], \varepsilon_{[\mathbf{x}]}, \varepsilon_{[\mathbf{p}]}$ )	
<b>saída</b> ( $\mathbf{Q}'_{[\mathbf{x}]}, \mathbf{Q}_{[\mathbf{p}]}$ )	
1	inicialize $\mathbf{Q}'_{[\mathbf{x}]}$ com $\emptyset$ , $\mathbf{Q}_{[\mathbf{x}]}$ com $[\mathbf{X}]$ e $\mathbf{Q}_{[\mathbf{p}]}$ com $[\mathbf{P}]$
2	gerar subpavimentos(entrada: $[\mathbf{P}], \varepsilon_{[\mathbf{p}]}$ ; saída: $\mathbf{Q}_{[\mathbf{p}]}$ ) (Tab.3.5)
3	se $\mathbf{Q}_{[\mathbf{x}]}$ está vazia siga para 8
4	retire $[\mathbf{x}]$ de $\mathbf{Q}_{[\mathbf{x}]}$
5	se $[\mathbf{g}]([\mathbf{x}], \mathbf{Q}_{[\mathbf{p}]}) \subset [0, \infty]$ então siga para 3
6	se $[\mathbf{g}]([\mathbf{x}], \mathbf{Q}_{[\mathbf{p}]}) \subseteq [-\infty, 0]$ então insira $[\mathbf{x}]$ em $\mathbf{Q}_{[\mathbf{x}]}$ e siga para 3
7	se $w([\mathbf{x}]) \geq \varepsilon_{[\mathbf{x}]}$ , bissecte $[\mathbf{x}]$ , insira as duas caixas resultantes em $\mathbf{Q}'_{[\mathbf{x}]}$ e siga para 3
8	retorne $\mathbf{Q}'_{[\mathbf{x}]}$ e $\mathbf{Q}_{[\mathbf{p}]}$

**Exemplo 14 (Tratamento de restrições)** Considere o espaço de busca  $[\mathbf{X}] = [-4, 4] \times [-3, 3]$  e o seguinte conjunto de funções de inclusão de restrições

$$\begin{aligned} [g_1] &: -[x_1]^2 - [x_2]^2 + 1 \subseteq [-\infty, 0] \\ [g_2] &: [x_2] - [x_1] - 3 \subseteq [-\infty, 0] \\ [g_3] &: -[x_2] + [x_1] - 3 \subseteq [-\infty, 0] \\ [g_4] &: [x_2] + [x_1] - 3 \subseteq [-\infty, 0] \\ [g_5] &: -[x_2] - [x_1] - 3 \subseteq [-\infty, 0]. \end{aligned} \quad (3.31)$$

O parâmetro de incerteza foi omitido por motivos de simplicidade. O espaço viável  $[\mathbf{X}]'$  é apresentado na Fig.3.5.

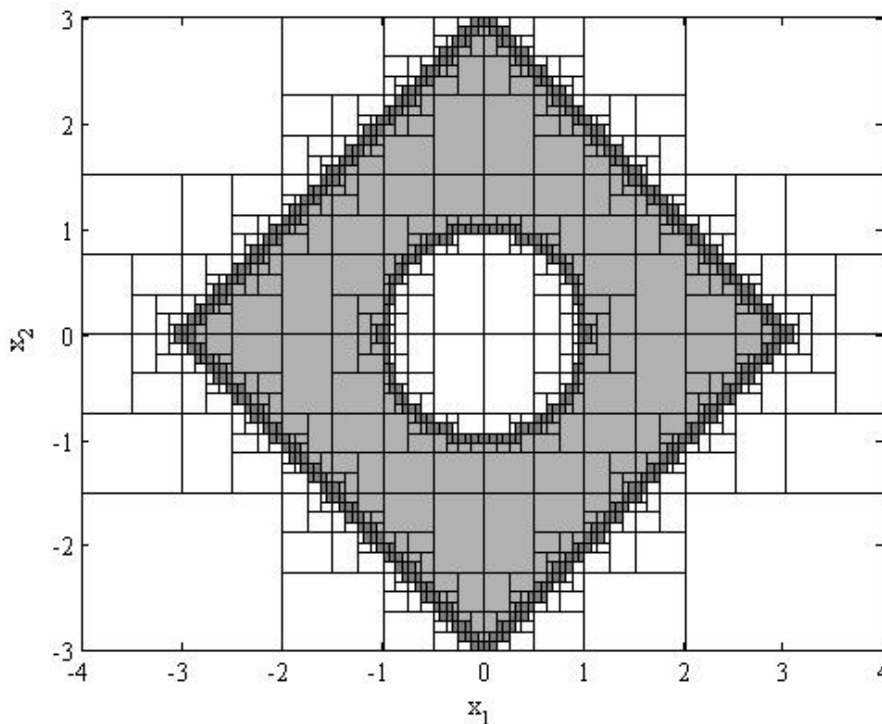


Fig. 3.5: O cinza claro representa o conjunto viável; o cinza escuro os intervalos de fronteira; e em branco os intervalos não viáveis. A figura foi obtida usando-se o algoritmo SIVIA (veja 2.3.12), com parâmetro de precisão  $\varepsilon = 0.05$ , para resolver as restrições propostas.

**Exclusão de regiões não robustas:** No algoritmo [I]RMOA II, os pontos ideais são armazenados na fila  $\mathbf{Q}_{\mathbf{Y}^*}$ . Agora considere qualquer  $[\mathbf{x}]$ , avaliado pela função objetivo  $[\mathbf{f}]([\mathbf{x}], [\mathbf{p}])$ , sendo  $[\mathbf{p}] \in \mathbf{Q}_{[\mathbf{p}]}$  como anteriormente. A função de inclusão mínima  $[\mathbf{f}]_{[\mathbf{x}]}^*$  é obtida conforme (3.28) e (3.29). Desta forma,

$$[\mathbf{x}] \text{ é descartado} \Leftrightarrow \exists \mathbf{y}^* \in \mathbf{Q}_{\mathbf{Y}^*}, \mathbf{y}^* \preceq [\mathbf{f}]_{[\mathbf{x}]}^*. \quad (3.32)$$

Portanto, a existência de pelo menos um elemento em  $\mathbf{Q}_{\mathbf{Y}^*}$  menor que  $[\mathbf{f}]_{[\mathbf{x}]}^*$ , é suficiente para excluir  $[\mathbf{x}]$  sem risco de se perder nenhuma solução robusta.

### 3.5.2 [I]RMOA II

O algoritmo [I]RMOA II é apresentado na Tab.3.7. As entradas são as caixas  $[\mathbf{X}]$  e  $[\mathbf{P}]$  com seus respectivos parâmetros de precisão  $\varepsilon_{[\mathbf{x}]}$  e  $\varepsilon_{[\mathbf{p}]}$ . O tratamento de restrições é realizado segundo a Tab.3.6, as regiões viáveis são armazenadas em  $\mathbf{Q}_{[\mathbf{x}]}$ . Enquanto  $\mathbf{Q}_{[\mathbf{x}]}$  permanecer não vazio, o processo de encontrar soluções robustas continua ativo. A primeira caixa  $[\mathbf{x}]$  é retirada de  $\mathbf{Q}_{[\mathbf{x}]}$ . Os próximos passos são computar a função de inclusão mínima  $[\mathbf{f}]_{[\mathbf{x}]}^*$  e comparar com os elementos na fronteira robusta em  $\mathbf{Q}_{\mathbf{Y}^*}$ . Se existe pelo menos um ponto em  $\mathbf{Q}_{\mathbf{Y}^*}$  menor que  $[\mathbf{f}]_{[\mathbf{x}]}^*$  então  $[\mathbf{x}]$  é descartado e retorna-se ao passo 3; senão  $\mathbf{c}([\mathbf{x}])$  é avaliado por  $\text{MIF}(\mathbf{c}([\mathbf{x}]), \mathbf{Q}_{[\mathbf{p}]})$  e a saída é armazenada  $[\mathbf{f}]_{\mathbf{x}}^*$ . Uma vez mais, se existe elementos em  $\mathbf{Q}_{\mathbf{Y}^*}$  menores que  $[\mathbf{f}]_{\mathbf{x}}^*$ , então conclui-se que  $\mathbf{c}([\mathbf{x}])$  não é solução robusta. Agora, se  $w([\mathbf{x}]) > \varepsilon_{[\mathbf{x}]}$  então  $[\mathbf{x}]$  é bissectado e os resultados inseridos em  $\mathbf{Q}_{[\mathbf{x}]}$ , e o laço é reiniciado no passo 3. Por outro lado,  $[\mathbf{x}]$  é inserido em  $\mathbf{Q}_{[\mathbf{x}]}$  e  $[\mathbf{f}]_{[\mathbf{x}]}^*$  em  $\mathbf{Q}_{[\mathbf{y}]}$ , e o algoritmo prossegue ao passo 10. Se não existe pontos menores que  $[\mathbf{f}]_{\mathbf{x}}^*$ , então o  $[\mathbf{f}]_{\mathbf{x}}^*$  faz parte da fronteira de Pareto robusta atual e é inserido em  $\mathbf{Q}_{\mathbf{Y}^*}$ , e  $\mathbf{c}([\mathbf{x}])$  em  $\mathbf{Q}_{\mathbf{X}^*}$ . O processo cíclico se mantém até que todas as caixas factíveis  $[\mathbf{x}]$  sejam investigadas. Ao final, é esperado que  $\mathbf{Q}_{[\mathbf{x}]}$  envelope todas as soluções do problema robusto e que  $\mathbf{Q}_{\mathbf{X}^*}$  e  $\mathbf{Q}_{\mathbf{Y}^*}$  contenham estimativas satisfatórias do conjunto solução e da fronteira robusta respectivamente. Como mostrado no Exemplo 13, talvez as supostas soluções de  $\mathbf{Q}_{\mathbf{Y}^*}$  não pertençam a  $\mathbf{f}(\mathbf{X}', \mathbf{P})$ . Portanto, outro método deve ser executado para validar os pontos em  $\mathbf{Q}_{\mathbf{Y}^*}$ , por exemplo o PROJ2D (Dao [27]) que computa projeção de conjuntos. A projeção de conjuntos está definida em (2.26) e a Fig. 2.3 ilustra um exemplo de projeção.

### 3.6 Algoritmo Evolucionário Intervalar Multi-Objetivo Robusto

Os algoritmos evolucionários utilizam processos estocásticos para guiar a busca pelas soluções ótimas. Contrariamente, algoritmos de otimização intervalares utilizam geralmente processos determinísticos. Considerando o modelo de otimização robusta descrito pelas expressões (3.9)-(3.18), apresenta-se nesta seção um algoritmo que mescla conceitos evolucionários e intervalares. Enquanto que a aritmética e algumas funções intervalares são empregadas para computar as incertezas sobre as soluções candidatas, os mecanismos evolucionários como seleção, recombinação e diversidade conduzem o método à convergência para a fronteira robusta. Denominou-se este algoritmo de [I]RMOEA (Interval Robust Multi-Objective Evolutionary Algorithm).

Tab. 3.7: Algoritmo [I]RMOA II.

	<b>entrada</b> ( $[\mathbf{X}], [\mathbf{P}], \varepsilon_{[x]}, \varepsilon_{[p]}$ )
	<b>saída</b> ( $\mathbf{Q}_{[x]}, \mathbf{Q}_{\mathbf{X}^*}, \mathbf{Q}_{\mathbf{Y}^*}$ )
1	inicialize $\mathbf{Q}_{\mathbf{X}^*}, \mathbf{Q}_{\mathbf{Y}^*}, \mathbf{Q}_{[y]}$ com $\emptyset$ .
2	tratar restrições(entrada: $[\mathbf{X}], [\mathbf{P}], \varepsilon_{[x]}, \varepsilon_{[p]}$ ; saída: $\mathbf{Q}_{[x]}, \mathbf{Q}_{[p]}$ ) (Tab.3.6)
3	se $\mathbf{Q}_{[x]}$ é vazio, então vá para 13; senão retire $[x]$ de $\mathbf{Q}_{[x]}$
4	chame MIF( $[x], \mathbf{Q}_{[p]}$ ), insira seu resultado em $[f]_{[x]}^*$
5	se existe um $y$ em $\mathbf{Q}_{\mathbf{Y}^*}$ tal que $y \preceq [f]_{[x]}^*$ então vá para 3
6	chame MIF( $c([x]), \mathbf{Q}_{[p]}$ ), insira seu resultado em $[f]_x^*$
7	se existe um $y$ em $\mathbf{Q}_{\mathbf{Y}^*}$ tal que $y \preceq \overline{[f]}_x^*$ ,
8	se $w([x]) > \varepsilon_{[x]}$ , então bissecte $[x]$ , insira as duas caixas resultantes em $\mathbf{Q}_{[x]}$ e siga para 3; senão insira $[f]_{[x]}^*$ em $\mathbf{Q}_{[y]}$ e $[x]$ em $\mathbf{Q}_{[x]}$
9	descarte todo $y$ em $\mathbf{Q}_{\mathbf{Y}^*}$ tal que $\overline{[f]}_x^* \preceq y$ e seus respectivos elementos em $\mathbf{Q}_{\mathbf{X}^*}$
10	insira $\overline{[f]}_x^*$ em $\mathbf{Q}_{\mathbf{Y}^*}$ e insira $c([x])$ em $\mathbf{Q}_{\mathbf{X}^*}$
11	descarte todo $[y]$ em $\mathbf{Q}_{[y]}$ tal que $[f]_x^* \preceq [y]$ e suas respectivas caixas em $\mathbf{Q}_{[x]}$
12	siga para o passo 3
13	retorne $\mathbf{Q}_{[x]}, \mathbf{Q}_{\mathbf{X}^*}$ e $\mathbf{Q}_{\mathbf{Y}^*}$

### 3.6.1 Preliminares

**Formação da população:** O parâmetro de busca  $\mathbf{x}^i$  corresponde à figura de um dado indivíduo  $i$  no [I]RMOEA, sendo formado pela concatenação de vetores pertencentes ao espaço de busca  $[\mathbf{X}] \in \mathbb{IR}^{n_x}$ . Algebricamente,

$$\mathbf{x}^i = x_1^i \times x_2^i \times \dots \times x_{n_x}^i, \quad \mathbf{x}^i \in [\mathbf{X}], \quad (3.33)$$

sendo  $i$  o identificador individual dentre os  $n_{pop}$  indivíduos da população. Na seqüência, numa geração  $t$ , a população é dada por

$$\mathbf{pop}_t = \{\mathbf{x}^1, \dots, \mathbf{x}^i, \dots, \mathbf{x}^{n_{pop}}\}, \quad (3.34)$$

sendo  $\mathbf{pop}_t(i)$  a forma de acesso ao  $i$ -ésimo membro.

**Avaliação da população:** A cada geração  $t$ , o desempenho da população  $\mathbf{des}_t$  frente às incertezas do sistema é representado por  $[f](\mathbf{pop}_t, [\mathbf{P}])$ . A

computação de  $[\mathbf{f}]$  é feita utilizando-se de métodos intervalares, similarmente ao [I]RMOA II conforme mostrado a seguir. Em primeiro lugar, assumamos que  $\mathbf{Q}_{[\mathbf{p}]}$  armazene todos os  $n_{sp}$  subpavimentos de  $[\mathbf{P}]$ , como já descrito anteriormente. Em segundo lugar, o desempenho de cada indivíduo  $\mathbf{x}^i$  é um envelope sobre uma região que considera a computação dos subpavimentos do espaço de incerteza sobre  $\mathbf{x}^i$ . Finalmente,  $\mathbf{des}_t$  é definido como o ponto ideal de maximização de cada  $[\mathbf{f}]_{\mathbf{x}^i}^*$ , ou seja

$$\mathbf{des}_t = \{[\mathbf{f}]_{\mathbf{x}^1}^*, \dots, [\mathbf{f}]_{\mathbf{x}^i}^*, \dots, [\mathbf{f}]_{\mathbf{x}^{n_{pop}}}^*\}. \quad (3.35)$$

O acesso ao  $i$ -ésimo membro é feito de forma análoga a  $\mathbf{pop}_t$ . O algoritmo da Tab.3.8 apresenta como é calculado o desempenho da população.

Tab. 3.8: Computação de  $[\mathbf{f}](\mathbf{pop}_t, [\mathbf{P}])$ .

<b>entrada</b> ( $\mathbf{pop}_t, \mathbf{Q}_{[\mathbf{p}]}$ )	%% denote $[\mathbf{P}]$ por $\mathbf{Q}_{[\mathbf{p}]}$
<b>saída</b> ( $\mathbf{des}_t$ )	
1	para $i = 1$ até $n_{pop}$ faça
2	chame $\text{MIF}(\mathbf{pop}_t(i), \mathbf{Q}_{[\mathbf{p}]})$ , insira seu resultado em $[\mathbf{f}]_{\mathbf{x}^i}^*$
3	fim para
4	armazene $\{[\mathbf{f}]_{\mathbf{x}^1}^*, \dots, [\mathbf{f}]_{\mathbf{x}^i}^*, \dots, [\mathbf{f}]_{\mathbf{x}^{n_{pop}}}^*\}$ em $\mathbf{des}_t$
5	retorne $\mathbf{des}_t$

**Tratamento de restrições:** A eliminação das regiões não viáveis ocorre de maneira análoga àquela apresentada na Subseção 3.5.1, pois o [I]RMOEA também realiza os testes para vetores reais. Assim, o teste de inclusão que representa as funções de restrição torna-se

$$[t](\mathbf{x}) = \begin{cases} 1, & \text{caso } [\mathbf{g}](\mathbf{x}, [\mathbf{p}]) \subseteq [-\infty, 0], \\ 0, & \text{caso } [\mathbf{g}](\mathbf{x}, [\mathbf{p}]) \subset [0, \infty], \\ [0, 1], & \text{por outro lado.} \end{cases} \quad (3.36)$$

**Método de seleção:** A seleção dos indivíduos para o cruzamento é realizada de acordo com o método Roleta, sendo no caso mono-objetivo, o percentual de escolha do indivíduo dependente do desempenho perante a população. Aqui, o método Roleta foi adaptado para o caso multi-objetivo avaliando o desempenho com o auxílio da técnica de nicho baseada em intervalos e da posição individual nas fronteiras de não dominância da geração atual. A técnica de nicho é descrita na próxima seção, mas para o momento, assumamos que

ela disponibilize informações sobre os nichos formados e os  $n_{nicho}$  indivíduos neles contidos. Então o desempenho de um indivíduo  $i$  segundo o nicho que ele participa é  $f_{nicho}^i = n_{nicho}$ . Suponha também que todos os indivíduos tenham sido classificados em fronteiras. O desempenho  $f_{front}^i$  é fixado em 1 caso o indivíduo  $i$  pertença à primeira fronteira, 2 para a segunda fronteira, etc. O resultado é o *desempenho útil*  $\mathbf{des}'_t(i) : \mathbb{R}$  medido como

$$\mathbf{des}'_t(i) = \frac{1}{f_{nicho}^i} + \frac{1}{f_{front}^i}. \quad (3.37)$$

O método da Roleta usa  $\mathbf{des}'_t(i) / \sum_{j=1}^{n_{pop}} \mathbf{des}'_t(j)$  para computar a probabilidade de cada indivíduo  $i$  ser escolhido. O método Roleta permite que alguns indivíduos sejam escolhidos várias vezes e outros nenhuma. Logo, considerando a transição da geração  $t$  para geração  $t + 1$ , é possível a perda de soluções pertencentes à fronteira eficiente. Por isso, tornou-se necessário aplicar uma técnica de elitismo para assegurar a permanência dos indivíduos considerados melhores que aqueles gerados na geração vigente.

**Método de elitismo:** Os métodos de elitismo têm a finalidade de impedir que soluções avaliadas como “melhores” ou “não piores” deixem de participar da próxima geração. A estratégia elitista utilizada aqui foi baseada na estratégia empregada no NSGA II (Deb [32]). Em poucas palavras, o método une a população na geração corrente  $\mathbf{pop}_t$  com a população recém surgida da aplicação dos operadores genéticos  $\mathbf{pop}_{aux}$ . Em seguida, compara todos os integrantes, separando os primeiros  $n_{pop}$  indivíduos da primeira fronteira para compor a nova população  $\mathbf{pop}_{t+1}$ . Enquanto o número de integrantes da primeira fronteira for menor que  $n_{pop}$  a técnica de nicho do [I]RMOEA é exatamente a mesma do NSGA II. Entretanto, quando a união de  $\mathbf{pop}_t$  com  $\mathbf{pop}_{aux}$  resultar em uma população na primeira fronteira maior que  $n_{pop}$  então a técnica de nicho novamente é acionada, e a população na primeira fronteira é dividida em nichos. Os nichos são percorridos, e a cada visita um de seus integrantes é selecionado aleatoriamente e retirado para compor a nova população. O processo continua até a seleção dos  $n_{pop}$  indivíduos.

### 3.6.2 Técnica de Nicho por Bissecção Intervalar

As técnicas de nicho foram desenvolvidas para preservar diversidade. Elas degradam a função objetivo conforme a distância entre as soluções, medida por uma *função de partilha*. Se há comparações entre todas as  $n_{pop}$  soluções com  $n_f$  objetivos é fácil perceber que o custo computacional é elevado. O custo é ainda mais crítico para algoritmos que classificam as fronteiras. Segundo Deb *et al.* [34], enquanto MOEAs desta natureza despendem esforço

na ordem de  $O(n_f * n_{pop}^3)$ , seu NSGA II gastava apenas  $O(n_f * n_{pop}^2)$ . Nesta seção é proposto uma técnica de nicho em que o custo não depende de comparações entre as  $n_{pop}$  soluções e os  $n_f$  objetivos, é a técnica de *Nichos por Bissecção Intervalar*.

Em poucas palavras, a técnica NB[I] (Nichos por Bissecção Intervalar) bissecta o espaço dos objetivos diversas vezes formando os nichos. A cada bissecção o volume dos nichos cai pela metade, e as soluções candidatas são classificadas nas sub-regiões acima e abaixo do ponto de corte. O resultado final fica disponível para o processo de seleção, onde as soluções têm seus desempenhos penalizados segundo o número de indivíduos de seu nicho. A técnica de nicho NB[I] é pontual podendo ser utilizada diretamente em algoritmos evolucionários tradicionais ou indiretamente nos algoritmos intervalares utilizando-se algum ponto de referência de intervalos para a classificação de pertinência aos nichos. Aqui, decidiu-se pela descrição do método conforme o emprego no algoritmo evolucionário-intervalar e utilizou-se o ponto ideal para maximização como o ponto de referência do intervalo.

A descrição da técnica de nichos, inicia-se assumindo uma população com  $n_{pop}$  indivíduos  $\mathbf{x}^i$ . Na iteração  $t$ , considere que o desempenho  $\mathbf{des}_t$  de cada  $\mathbf{x}^i$  seja calculado conforme mostrado na Tab.3.8. Então, é sempre verdadeira a seguinte inclusão

$$\overline{\mathbf{f}}_{\mathbf{x}^i}^* \subseteq [\mathbf{F}]_0, \quad [\mathbf{F}]_0 = [\underline{F}_1, \overline{F}_1] \times \dots \times [\underline{F}_j, \overline{F}_j] \times \dots \times [\underline{F}_{n_f}, \overline{F}_{n_f}], \quad (3.38)$$

sendo

$$\underline{F}_j = \min_{i=1, \dots, n_{pop}} \overline{[f_j]_{\mathbf{x}^i}}^* \quad \text{e} \quad \overline{F}_j = \max_{i=1, \dots, n_{pop}} \overline{[f_j]_{\mathbf{x}^i}}^*. \quad (3.39)$$

Em toda geração,  $[\mathbf{F}]_0$  representa o nicho inicial contendo toda a população. O próximo passo é bissecionar  $[\mathbf{F}]_0$  em uma de suas  $n_f$  funções objetivo, escolhida aleatoriamente. Por exemplo, assuma que a função objetivo  $j$  seja sorteada. O resultado são dois novos nichos  $[\mathbf{F}]_1$  e  $[\mathbf{F}]_2$  tais que

$$[\mathbf{F}]_1 = [F_1] \times \dots \times [\underline{F}_j, \frac{F_j + \overline{F}_j}{2}] \times \dots \times [F_{n_f}], \quad (3.40)$$

$$[\mathbf{F}]_2 = [F_1] \times \dots \times [\frac{F_j + \overline{F}_j}{2}, \overline{F}_j] \times \dots \times [F_{n_f}]. \quad (3.41)$$

Assim, a classificação de cada  $\mathbf{x}^i$  é dada por

$$\begin{cases} \overline{\mathbf{f}}_{\mathbf{x}^i}^* \in [\mathbf{F}]_1, & \text{caso } \overline{[f_j]_{\mathbf{x}^i}}^* \leq \frac{F_j + \overline{F}_j}{2}, \\ \overline{\mathbf{f}}_{\mathbf{x}^i}^* \in [\mathbf{F}]_2, & \text{por outro lado.} \end{cases} \quad (3.42)$$

Após o processo de classificação, se algum nicho for vazio, ele é eliminado. O processo de bissecção continua enquanto algum critério de parada seja atingido, como o *número máximo de rodadas de bissecção*  $r_{bis}^{max}$ . A Fig. 3.6 mostra as quatro primeiras bissecções num problema genérico com dois objetivos.

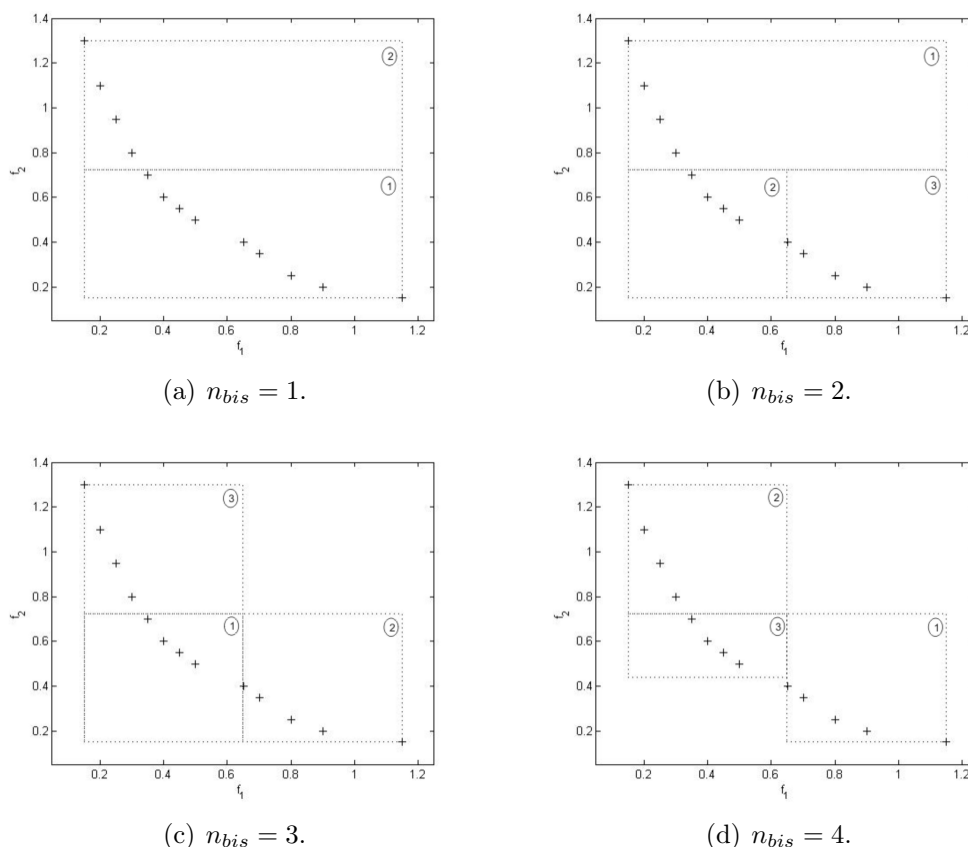


Fig. 3.6: NB[I]: primeiras bissecções sobre uma população com 13 indivíduos. Assuma que “+” denote  $\overline{[\mathbf{f}]_{\mathbf{x}^i}}$ . Observe que entre (c) e (d) há exclusão das regiões que não contêm solução. No canto direito superior de cada região há indicação da ordem de classificação das caixas para bissecção.

Analisando o grupo de gráficos da Fig. 3.6, percebe-se que o processo de agrupar pontos vizinhos é simples. Decidiu-se por implementá-lo usando a estrutura de dados baseada em listas encadeadas FIFO para armazenar os nichos e os índices dos indivíduos que cada um contém. O algoritmo da Tab.3.9. apresenta a técnica NBI[I] e a Tab.3.10 descreve a simbologia empregada no algoritmo.

Tab. 3.9: Algoritmo NB[I].

	<b>entrada</b> ( $\mathbf{des}_t, r_{bis}^{max}$ )
	<b>saída</b> ( $\mathbf{Q}_{[F]}, \mathbf{Q}_{ind}$ )
1	inicialize $r_{bis}$ com 0, $\mathbf{Q}_{[F]}$ com $\emptyset$ e $\mathbf{Q}_{ind}\{1\}$ com os índices $1, \dots, n_{pop}$
2	usando $\mathbf{des}_t$ em (3.38) e (3.39) compute $[\mathbf{F}]_0$ e insira em $\mathbf{Q}_{[F]}$
3	se $r_{bis} > r_{bis}^{max}$ siga para 14
4	adicione 1 a $r_{bis}$
5	retire o primeiro elemento de $\mathbf{Q}_{[F]}$ e armazene em $[\mathbf{F}]$
6	retire os índices de $\mathbf{Q}_{ind}\{1\}$ e armazene em $\mathbf{Q}_{ind}^{[F]}$
7	bissecte $[\mathbf{F}]$ e armazene o resultado em $[\mathbf{F}]_1$ e $[\mathbf{F}]_2$
8	usando (3.42), classifique os índices em $\mathbf{Q}_{ind}^{[F]}$ e armazene o resultado em $\mathbf{Q}_{ind}^{[F]1}$ e $\mathbf{Q}_{ind}^{[F]2}$
9	se $\mathbf{Q}_{ind}^{[F]1} = \emptyset$ siga para 11
10	insira $[\mathbf{F}]_1$ em $\mathbf{Q}_{[F]}$ e $\mathbf{Q}_{ind}^{[F]1}$ em $\mathbf{Q}_{ind}\{(\text{última posição}) + 1\}$
11	se $\mathbf{Q}_{ind}^{[F]2} = \emptyset$ siga para 3
12	insira $[\mathbf{F}]_2$ em $\mathbf{Q}_{[F]}$ e $\mathbf{Q}_{ind}^{[F]2}$ em $\mathbf{Q}_{ind}\{(\text{última posição}) + 1\}$
13	siga para 3
14	retorne $\mathbf{Q}_{[F]}$ e $\mathbf{Q}_{ind}$

Tab. 3.10: Símbolos no algoritmo NB[I].

Símbolos	Descrição	Conteúdo	Tipo
$n_{pop}$	tamanho da população		$\mathbb{Z}$
$r_{bis}$	contador de rodadas de bissecção		$\mathbb{Z}$
$r_{bis}^{max}$	número máximo de rodadas de bissecção		$\mathbb{Z}$
$\mathbf{des}_t$	desempenho da população		$\mathbb{R}^{n_f}$
$[\mathbf{F}]$	caixa do nicho corrente		$\mathbb{IR}^{n_f}$
$[\mathbf{F}]_1, [\mathbf{F}]_2$	caixas dos nichos recém bisseccionados		$\mathbb{IR}^{n_f}$
$\mathbf{Q}_{[F]}$	Lista FIFO com nichos ativos		Fila: $\mathbb{IR}^{n_f}$
$\mathbf{Q}_{ind}^{[F]}$	Lista FIFO com índices dos indivíduos de $\mathbf{Q}_{[F]}\{1\}$		Fila: $\mathbb{R}$
$\mathbf{Q}_{ind}^{[F]1}, \mathbf{Q}_{ind}^{[F]2}$	Lista FIFO com índices de nichos correntes		Fila: $\mathbb{IR}^{n_f}$
$\mathbf{Q}_{ind}\{.\}$	Lista FIFO com índices dos indivíduos de $\mathbf{Q}_{[F]}\{.\}$		Fila: $\mathbb{R}$

Como qualquer método de nicho, NB[I] também contém parâmetros críticos que interferem na qualidade da formação dos nichos. Estes parâmetros são ligados ao critério de parada escolhido. Por exemplo, a Fig. 3.7 mostra

os nichos resultantes da técnica NB[I] para diferentes valores para  $r_{bis}^{max}$ . Se muitas bissecções ocorrerem, os nichos limitam apenas um ponto isolado, veja Fig. 3.7(a). Por outro lado, se poucos cortes forem feitos, a técnica de nicho falha em agrupar somente soluções vizinhas. Considerando o caso

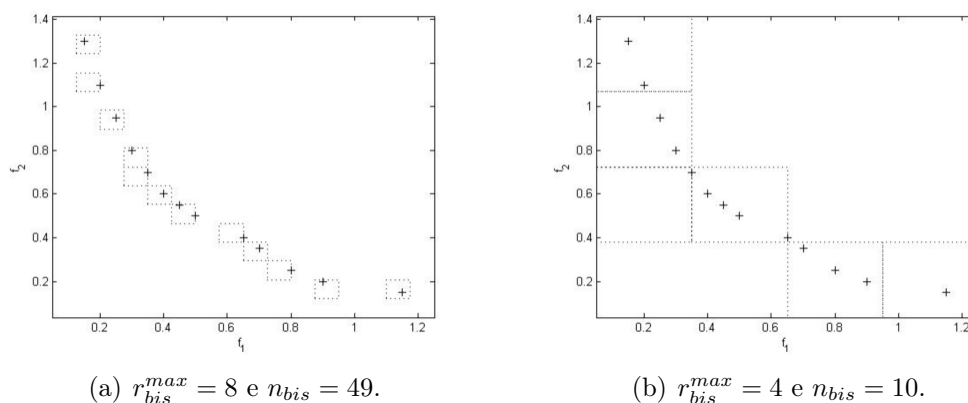


Fig. 3.7: Técnica NB[I] - problemas com bissecções. Em (a), NB[I] falha pois os nichos contêm apenas um indivíduo cada. Em (b), indivíduos próximos se situam em nichos diferentes.

onde o critério baseia-se em  $r_{bis}^{max}$ , percebe-se que há necessidade de descobrir um valor satisfatório de bissecções que permita explorar a informação de proximidade entre os indivíduos. Uma solução seria deixar o valor de  $r_{bis}^{max}$  variar, por exemplo, segundo um valor de uma *densidade populacional*. Mas isso significa acréscimo de mais um parâmetro crítico. A Fig. 3.7(b) mostra outro aspecto importante: soluções próximas que estão em nichos diferentes. Este problema pode ser amenizado executando NB[I] mais vezes, porém utilizando-se de espaços de busca iniciais diferentes. No final, o desempenho seria calculado empregando-se uma média aritmética entre as densidades populacionais que dada solução participa.

O algoritmo NB[I] pode ser facilmente adaptado para outros critérios de parada, como por exemplo, o NB[I] pode encerrar as bissecções impedindo que o volume do nicho seja menor que o volume de nicho mínimo, fixado previamente.

### 3.6.3 [I]RMOEA

O algoritmo [I]RMOEA apresentado na Tab.3.11 tem estrutura semelhante a de outros algoritmos genéticos multi-objetivo, no entanto, a estrutura de dados e as operações genéticas são baseadas em parâmetros reais e intervalares. A entrada contém todos os parâmetros utilizados no [I]RMOEA, cuja descrição está mostrada na Tab.3.12. Em 2 as soluções são testadas quanto à

Tab. 3.11: Algoritmo [I]RMOEA.

	<b>entrada</b> ( $n_{bits}, n_f, n_{pop}, n_{gen}, p_c, p_m, [\mathbf{X}], [\mathbf{P}], \varepsilon_{[x]}, \varepsilon_{[p]}$ )
	<b>saída</b> ( $\mathbf{pop}_t, \mathbf{des}_t$ )
1	inicialize $t$ com 0
2	tratar restrições(entrada: $[\mathbf{X}], [\mathbf{P}], \varepsilon_{[x]}, \varepsilon_{[p]}$ ; saída: $\mathbf{Q}_{[x]}, \mathbf{Q}_{[p]}$ ) (Tab.3.6)
3	inicialize $\mathbf{pop}_t$ com $n_{bits}, n_f, n_{pop}$ e $\mathbf{Q}_{[x]}$
4	compute o desempenho $[\mathbf{f}](\mathbf{pop}_t, \mathbf{Q}_{[p]})$ e armazene-o em $\mathbf{des}_t$
5	separe as fronteiras e armazene o resultado em $f_{front}^i$
6	se $t \geq n_{gen}$ siga para 18
7	incremente 1 a $t$
8	chame NBI e armazene o resultado em $f_{nicho}^i$
9	utilize 3.37 para computar $\mathbf{des}'_t$
10	utilize $\mathbf{des}'_t$ para selecionar casais para <i>cruzamento</i>
11	realize <i>cruzamento</i> ; <i>mutação</i> segundo a probabilidade $p_m$ , armazene o resultado em $\mathbf{pop}_{aux}$
12	substitua todo indivíduo $\mathbf{x} \in \mathbf{pop}_{aux}$ tal que $[\mathbf{g}](\mathbf{x}, \mathbf{Q}_{[p]}) \subset [0, \infty]$ por outro gerado aleatoriamente de $\mathbf{Q}_{[x]}$
13	compute o desempenho $[\mathbf{f}](\mathbf{pop}_{aux}, \mathbf{Q}_{[p]})$ e armazene-o em $\mathbf{des}_{aux}$
14	armazene $\mathbf{pop}_{t-1} \cup \mathbf{pop}_{aux}$ em $\mathbf{pop}_{aux}$ ; $\mathbf{des}_{t-1} \cup \mathbf{des}_{aux}$ em $\mathbf{des}_{aux}$
15	separe as fronteiras e armazene o resultado em $f_{front}^i$
16	finalize a população para a próxima geração com os $n_{pop}$ primeiros indivíduos de $f_{front}^i$
17	siga para 6
18	retorne $\mathbf{pop}_t$ e $\mathbf{des}_t$

viabilidade; as não viáveis são substituídas. Em 3 a população é inicializada. Em 4, a população é submetida à avaliação de desempenho expressa pelas funções objetivo. O resultado é armazenado em  $\mathbf{des}_t$ . O desempenho  $\mathbf{des}_t$  é utilizado na separação de fronteiras no passo 5. O passo 6 marca o início do laço principal. O critério de parada foi definido como número máximo de gerações. Segundo sua conveniência, o decisor pode inserir outro critério. O contador de gerações é atualizado e em seguida, no passo 8 – 9 a técnica de nicho NB[I] é aplicada e o resultado  $f_{nicho}^i$  torna-se suporte para os métodos de seleção, cruzamento e mutação nas etapas subseqüentes (passos 9 – 11). Novos testes de restrição são realizados, assim como uma nova avaliação de desempenho (passos 12 – 13). Em 14 – 16, As populações  $\mathbf{pop}_{t-1}$  e  $\mathbf{pop}_{aux}$ ,

e os desempenhos  $\mathbf{des}_{t-1}$  e  $\mathbf{des}_{aux}$  são acoplados, e fronteiras robustas separadas pelo método do elitismo. Os primeiros  $n_{pop}$  elementos das primeiras fronteiras formam  $\mathbf{pop}_t$ . O restante dos indivíduos são excluídos. Claramente, o algoritmo descrito acima se assemelha a um MOEA típico, que não leva em conta o tratamento de incertezas. Contudo, as diferenças estão nos detalhes: a) as funções objetivo contêm o parâmetro de incerteza  $[\mathbf{P}]$ ; e b) os métodos intervalares são empregados para computar a incerteza e para classificar os indivíduos na técnica de nicho.

Tab. 3.12: Parâmetros do [I]RMOEA

Parâmetro	Descrição	Conteúdo	Tipo
$n_{bits}$	número de bits da cadeia binária		$\mathbb{Z}$
$n_{gen}$	número de gerações		$\mathbb{Z}$
$n_{pop}$	número de indivíduos da população		$\mathbb{Z}$
$n_f$	número de funções objetivo		$\mathbb{Z}$
$p_c$	probabilidade de cruzamento		$\mathbb{R}$
$p_m$	probabilidade de mutação		$\mathbb{R}$
$f_{front}^i$	desempenho do indivíduo $i$ segundo sua fronteira		$\mathbb{Z}$
$f_{nicho}^i$	desempenho do indivíduo $i$ segundo seu nicho		$\mathbb{Z}$
$\mathbf{pop}_t$	população na geração $t$		Vetor: $\mathbb{R}^{n_f}$
$\mathbf{des}_t$	desempenho de $\mathbf{pop}_t$		Vetor: $\mathbb{R}^{n_f}$
$\mathbf{pop}_{aux}$	população auxiliar		Vetor: $\mathbb{R}^{n_f}$
$\mathbf{des}_{aux}$	desempenho de $\mathbf{pop}_{aux}$		Vetor: $\mathbb{R}^{n_f}$

Como dito anteriormente, os algoritmos evolucionários buscam pelos pontos ótimos via mecanismos estocásticos. Portanto, o desempenho destes algoritmos está associado a escolha correta dos parâmetros de controle destes mecanismos.

### 3.7 Métricas para Medir Desempenho Revisadas

Na Seção 2.2.6 foram apresentadas algumas métricas utilizadas para testar desempenho dos MOEAs. Agora, apresenta-se a Métrica por Bissecção de Intervalar que é empregada para avaliar algoritmos propostos nesta Tese com relação à aproximação da fronteira robusta.

## 3.7.1 Métrica por Bissecção Intervalar - MB[I]

A métrica foi desenvolvida para qualificar a uniformidade de um conjunto de soluções não dominadas de um espaço bidimensional em relação ao conjunto Pareto  $\mathbf{Y}^*$ . Os casos n-dimensionais podem ser obtidos por analogia. Seguindo a idéia básica da técnica de nicho NB[I], esta métrica subdivide o espaço dos objetivos escolhendo, a cada iteração, a imagem de uma solução como ponto de bissecção para as duas novas regiões. As bissecções vão ocorrendo até que não haja mais soluções internas às regiões restantes dos processos de bissecção. A Fig. 3.8 ilustra quatro etapas do processo.

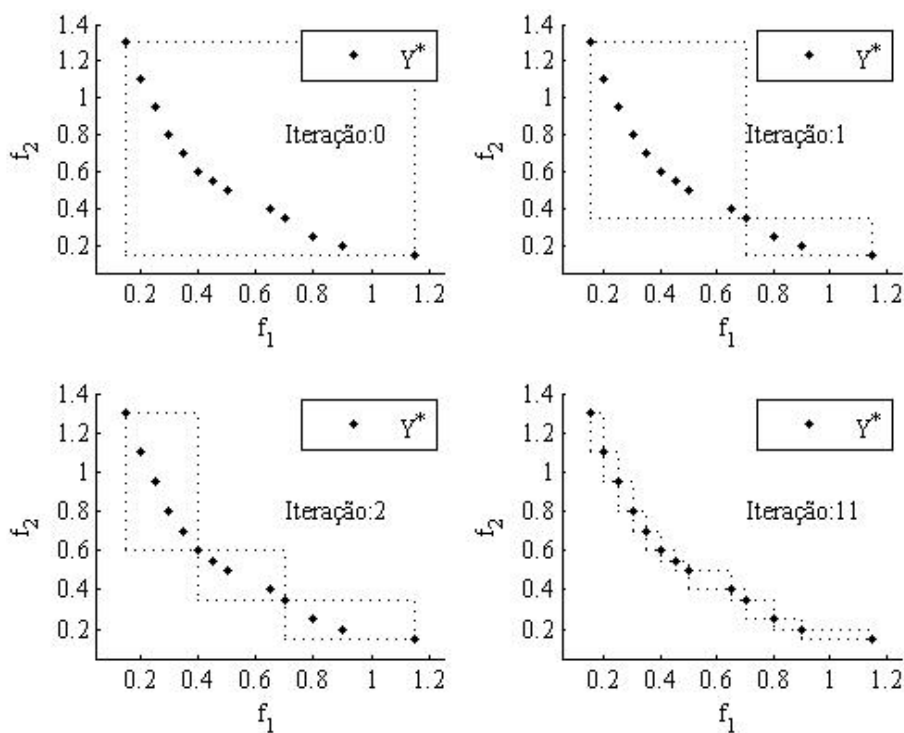


Fig. 3.8: Algumas iterações da métrica MB[I] sobre uma população de soluções não dominadas. Em cada iteração há exclusão das regiões que não contêm solução.

A descrição formal do método MB[I] inicia-se por considerar o desempenho  $\mathbf{des}_t$  de uma população de pontos em uma geração  $t$ , como

$$\mathbf{des}_t = \{ \mathbf{f}_{\mathbf{x}^1}, \dots, \mathbf{f}_{\mathbf{x}^i}, \dots, \mathbf{f}_{\mathbf{x}^{n_{pop}}} \}, \quad i = \{1, \dots, n_{pop}\} \quad (3.43)$$

sendo  $\mathbf{f}_{\mathbf{x}^i} \in \mathbb{R}^{n_f}$  e o acesso à  $j$ -ésima função objetivo do indivíduo  $\mathbf{x}^i$  dá-se por  $f_{j_{\mathbf{x}^i}}$ . Assim, para todo indivíduo  $\mathbf{f}_{\mathbf{x}^i}$  da população é sempre verdadeira

a seguinte inclusão

$$\mathbf{f}_{\mathbf{x}^i} \subseteq [\mathbf{F}]'_0, \quad [\mathbf{F}]'_0 = [\underline{F}'_1, \overline{F}'_1] \times \dots \times [\underline{F}'_j, \overline{F}'_j] \times \dots \times [\underline{F}'_{n_f}, \overline{F}'_{n_f}], \quad (3.44)$$

sendo

$$\underline{F}'_j = \min_{i=1, \dots, n_{pop}} f_{j_{\mathbf{x}^i}} \quad \text{e} \quad \overline{F}'_j = \max_{i=1, \dots, n_{pop}} f_{j_{\mathbf{x}^i}}. \quad (3.45)$$

Um exemplo da construção de  $[\mathbf{F}]'_0$  pode ser visto na Fig. 3.8, iteração 0. A caixa  $[\mathbf{F}]'_0$  inicializa a fila de caixas  $\mathbf{Q}_{[\mathbf{F}]}$ . O próximo passo é bisseccionar  $[\mathbf{F}]'_0$ , segundo um de seus pontos internos, ou seja quaisquer pontos exceto aqueles localdos nos vértices. Os índices dos indivíduos situados nos vértices são armazenados em  $\mathbf{Q}_{[\mathbf{F}]}^v$ , escolhas aleatórias são feitas: a) um índice  $k$  ( $k = \{1, \dots, n_{pop}\}$  e  $k \notin \mathbf{Q}_{[\mathbf{F}]}^v$ ) para o indivíduo de corte  $\mathbf{f}_{\mathbf{x}^k}$ ; e b) um índice  $j$  para a função objetivo do indivíduo  $k$  escolhido,  $f_{j_{\mathbf{x}^k}}$ . Após as escolhas, duas novas regiões são geradas  $[\mathbf{F}]_1$  e  $[\mathbf{F}]_2$  expressadas como

$$[\mathbf{F}]'_1 = [F_1] \times \dots \times [\underline{F}_j, f_{j_{\mathbf{x}^k}}] \times \dots \times [F_{n_f}], \quad (3.46)$$

$$[\mathbf{F}]'_2 = [F_1] \times \dots \times [f_{j_{\mathbf{x}^k}}, \overline{F}_j] \times \dots \times [F_{n_f}]. \quad (3.47)$$

Para melhor compreensão, veja Fig. 3.8, iteração 1. A lista de vértices  $\mathbf{Q}_{[\mathbf{F}]}^v$  deve ser atualizada após a bissecção incluindo o ponto de corte entre as novas caixas  $[\mathbf{F}]'_1$  e  $[\mathbf{F}]'_2$ . Em seguida, os indivíduos até então pertencentes a  $[\mathbf{F}]'_0$  devem ser classificados quanto ao novo espaço a que pertencem, com

$$\begin{cases} \mathbf{f}_{\mathbf{x}^i} \in [\mathbf{F}]_1, & \text{caso } f_{j_{\mathbf{x}^i}} \leq f_{j_{\mathbf{x}^k}}, \\ \mathbf{f}_{\mathbf{x}^i} \in [\mathbf{F}]_2, & \text{por outro lado.} \end{cases} \quad (3.48)$$

O processo de bissecção continua enquanto houver soluções internas às caixas intervalares, ou similarmente quando o número de elementos de  $\mathbf{Q}_{[\mathbf{F}]}$  for igual a  $n_{pop} - 1$ . Um exemplo de fim das bissecções é mostrado na Fig. 3.8, iteração 11. Depois de executar todas as bissecções, computam-se os volumes  $v([\mathbf{F}])$  das caixas intervalares restantes, para então calcular o desvio padrão  $\sigma_{v([\mathbf{F}])}$  do grupo. Esse desvio padrão  $\sigma_{v([\mathbf{F}])}$  corresponde ao valor quantitativo da métrica MB[I], ou formalmente

$$\text{MB}[\mathbf{I}] = \sigma_{v([\mathbf{F}])}. \quad (3.49)$$

Quanto menor  $\sigma_{v([\mathbf{F}])}$  mais uniformemente está distribuída a população. O algoritmo da Tab.3.13. apresenta a técnica MBI[I] e a Tab.3.14 descreve a simbologia empregada no algoritmo.

A aplicação da métrica MB[I] para qualificar a uniformidade de um conjunto de pontos não dominados só tem validade quando comparado com um

Tab. 3.13: Algoritmo MB[I].

	<b>entrada</b> ( $\mathbf{des}_t$ )
	<b>saída</b> (MB[I])
1	inicialize $\mathbf{Q}_{[\mathbf{F}]}$ e $\mathbf{Q}_{[\mathbf{F}]}^v$ com $\emptyset$ ; $\mathbf{Q}_{\text{ind}}$ com os índices $1, \dots, n_{pop}$
2	compute $[\mathbf{F}]'_0$ usando $\mathbf{des}_t$ em (3.44) e (3.45)
3	insira $[\mathbf{F}]'_0$ em $\mathbf{Q}_{[\mathbf{F}]}$
4	insira os índices dos pontos que definem os vértices de $[\mathbf{F}]'_0$ em $\mathbf{Q}_{[\mathbf{F}]}^v$
5	atualize $\mathbf{Q}_{\text{ind}}\{1\}$ com $\mathbf{Q}_{\text{ind}}\{1\} \setminus \mathbf{Q}_{[\mathbf{F}]}^v\{1\}$
6	se o número de elementos de $\mathbf{Q}_{[\mathbf{F}]}$ é igual a $n_{pop} - 1$ siga para 19
7	retire o primeiro elemento de $\mathbf{Q}_{[\mathbf{F}]}$ e armazene em $[\mathbf{F}]$
8	retire os índices de $\mathbf{Q}_{\text{ind}}\{1\}$ e armazene em $[\mathbf{F}]_{\text{ind}}$
9	se $[\mathbf{F}]_{\text{ind}} = \emptyset$ insira: $[\mathbf{F}]$ em $\mathbf{Q}_{[\mathbf{F}]}$ e $[\mathbf{F}]_{\text{ind}}$ em $\mathbf{Q}_{\text{ind}}$ ; siga para 6
10	sorteie um índice de $[\mathbf{F}]_{\text{ind}}$ e armazene em $k$
11	sorteie um objetivo de $\{1, \dots, n_f\}$ e armazene em $j$
12	bissecte $[\mathbf{F}]$ em $\mathbf{f}_{j \times k}$ e armazene os resultados em $[\mathbf{F}]'_1$ e $[\mathbf{F}]'_2$
13	classifique os indivíduos em $[\mathbf{F}]'_1$ e $[\mathbf{F}]'_2$ usando (3.48), armazenando seus índices em $[\mathbf{F}]'_{1 \text{ ind}}$ e/ou $[\mathbf{F}]'_{2 \text{ ind}}$
14	se $[\mathbf{F}]'_{1 \text{ ind}} = \emptyset$ siga para 16
15	insira $[\mathbf{F}]'_1$ em $\mathbf{Q}_{[\mathbf{F}]}$ e $[\mathbf{F}]'_{1 \text{ ind}}$ em $\mathbf{Q}_{\text{ind}}$
16	se $[\mathbf{F}]'_{2 \text{ ind}} = \emptyset$ siga para 6
17	insira $[\mathbf{F}]'_2$ em $\mathbf{Q}_{[\mathbf{F}]}$ e $[\mathbf{F}]'_{2 \text{ ind}}$ em $\mathbf{Q}_{\text{ind}}$ , siga para 3
18	siga para 6
19	compute os volumes $v(\mathbf{Q}_{[\mathbf{F}]} \{i\})$ , sendo $i = 1, \dots, n_{pop} - 1$
20	calcule o desvio padrão $\sigma_{v(\mathbf{Q}_{[\mathbf{F}]} \{i\})}$ e armazene o resultado em MB[I]
21	retorne MB[I]

valor de referência, como por exemplo, o valor da métrica sobre a fronteira de Pareto computada sob dada precisão, aqui designada por  $\text{MB}[\mathbf{I}]_{\text{front}}$ . Ainda, comparar o valor de MB[I] diretamente com  $\text{MB}[\mathbf{I}]_{\text{front}}$  pode não resultar em análise clara, por causa dos valores numéricos que MB[I] pode gerar. A situação se agrava quando há comparação simultânea da métrica para mais de uma simulação. Por tudo isso, torna-se importante desenvolver uma função de transformação para normalizar a métrica MB[I]. A métrica por bissecção intervalar normalizada  $\text{MB}[\mathbf{I}]_{\text{norm}}$  é descrita como

$$\text{MB}[\mathbf{I}]_{\text{norm}} = \frac{\text{MB}[\mathbf{I}] - \text{MB}[\mathbf{I}]_{\text{front}}}{\text{MB}[\mathbf{I}]_{\text{max}} - \text{MB}[\mathbf{I}]_{\text{front}}}, \quad \text{MB}[\mathbf{I}]_{\text{max}} \neq \text{MB}[\mathbf{I}]_{\text{front}}, \quad (3.50)$$

Tab. 3.14: Símbolos no algoritmo MB[I].

Símbolos	Descrição/Conteúdo	Tipo
$n_{pop}$	tamanho da população	$\mathbb{R}$
$n_{pop} - 1$	nº máximo de caixas intervalares entre soluções	$\mathbb{R}$
$\mathbf{des}_t$	desempenho da população	Vetor: $\mathbb{R}^{n_f}$
$\mathbf{f}_{\mathbf{x}^k}$	desempenho do indivíduo $k$	$\mathbb{R}^{n_f}$
$[\mathbf{F}]_0$	caixa intervalar inicial	$\mathbb{IR}^{n_f}$
$[\mathbf{F}]$	caixa intervalar corrente	$\mathbb{IR}^{n_f}$
$[\mathbf{F}]'_1, [\mathbf{F}]'_2$	caixas intervalares temporárias	$\mathbb{IR}^{n_f}$
$[\mathbf{F}]_{\text{ind}}$	índices dos indivíduos internos a $[\mathbf{F}]$	Vetor: $\mathbb{R}$
$[\mathbf{F}]'_{1 \text{ ind}}, [\mathbf{F}]'_{2 \text{ ind}}$	índices dos indivíduos internos a $[\mathbf{F}]'_1$ e $[\mathbf{F}]'_2$	Vetor: $\mathbb{R}$
$\mathbf{Q}_{[\mathbf{F}]}$	caixas intervalares	Fila: $\mathbb{IR}^{n_f}$
$\mathbf{Q}_{[\mathbf{F}]}^v$	índices dos indivíduos nos vértices	Fila: $\mathbb{R}$
$\mathbf{Q}_{\text{ind}}$	índices dos indivíduos internos a $\mathbf{Q}_{[\mathbf{F}]}$	Fila: $\mathbb{R}$
$v(\mathbf{Q}_{[\mathbf{F}]} \{i\})$	volume da caixa armazenada em $\mathbf{Q}_{[\mathbf{F}]} \{i\}$	$\mathbb{R}^{n_f}$
$\sigma_{v(\mathbf{Q}_{[\mathbf{F}]} \{i\})}$	desvio padrão dos volumes $v(\mathbf{Q}_{[\mathbf{F}]} \{i\})$	$\mathbb{R}$

sendo  $\text{MB}[\mathbf{I}]$  o valor da métrica para dada execução e  $\text{MB}[\mathbf{I}]_{\max}$  o valor máximo da métrica para todas execuções em um dado teste.

### 3.7.2 $\text{MB}[\mathbf{I}] \times$ Métricas Apresentadas

Após a apresentação da Métrica por Bisseção de Intervalos, é feita nesta subseção, uma análise numérica de todas as métricas apresentadas através do estudo de casos fictícios de fronteiras Pareto, que podem ser robustas ou não. Em seguida, é feita a justificativa das métricas escolhidas para avaliar o algoritmo evolucionário no próximo capítulo. Para melhor compreensão da análise numérica registra-se que: a) em todos os gráficos, existem dois conjuntos de pontos  $\mathbf{A}$  e  $\mathbf{B}$  e, ocasionalmente um terceiro conjunto  $C$ ; b) a fronteira de Pareto  $\mathbf{Y}^*$  não está explicitada para deixar mais claros os exemplos gráficos. Para o leitor que deseje examinar métricas, basta observar que  $\mathbf{Y}^*$  é formado por todos os pontos eficientes existentes no gráfico; c) os conjuntos avaliados são, dentro de seu grupo, não dominados; d) índices quando acrescentados denotam o conjunto sobre o qual a métrica foi calculada; e) os cálculos para obtenção dos resultados mostrados nas figuras foram arredondados com precisão de 0.001; f) não empregou-se métrica  $\text{MB}[\mathbf{I}]$  normalizada por causa da simplicidade dos testes, sendo utilizado o resultado direto do algoritmo da Tab.3.13; e g) durante a avaliação de caso de teste, as métricas são marcadas com o sinal “ $\surd$ ” para indicar a aprovação da métrica,

e com o “?” para indicar que a métrica não obteve resultado satisfatório para a avaliação dos conjuntos solução.

*Caso 1: dominância completa entre conjuntos solução*

No primeiro estudo, mostrado pela Fig. 3.9, é fácil notar que o conjunto **A** domina completamente **B**, e ainda que **A** é melhor distribuído que **B**.

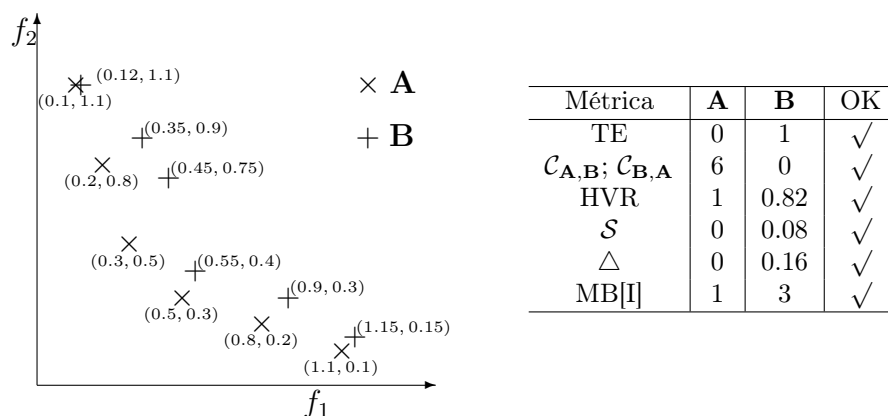


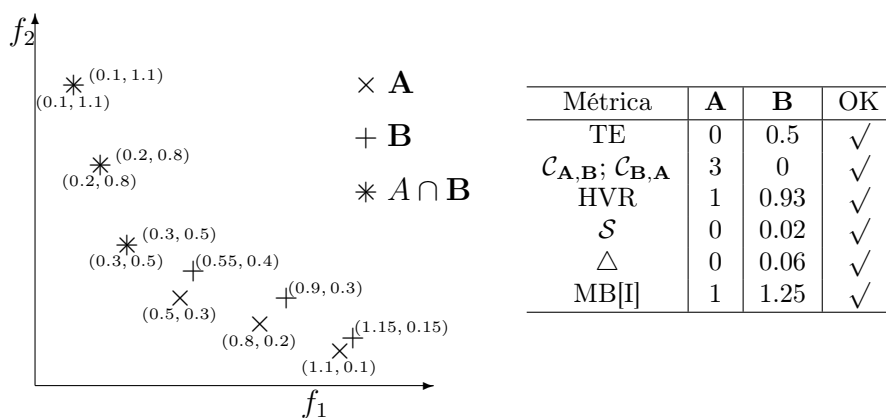
Fig. 3.9: Dominância completa do conjunto **A** em relação ao conjunto **B**.

Como pode ser visto, as métricas apontam **A** melhor que **B** e por isso todas são avaliadas positivamente. Cada componente de **B** é dominado por pelo menos um componente de **A**. Nota-se claramente que a métrica *TE* contabiliza a presença dos elementos de **A** em  $\mathbf{Y}^*$ . Entretanto, considere que os conjuntos **A** e **B** fossem translados de tal forma que mantivessem o mesmo posicionamento entre si, porém com nenhum ponto de **A** em  $\mathbf{Y}^*$ . Neste caso, a métrica não deixaria claro quem teria a melhor ou pior fronteira.

*Caso 2: dominância forte entre conjuntos solução*

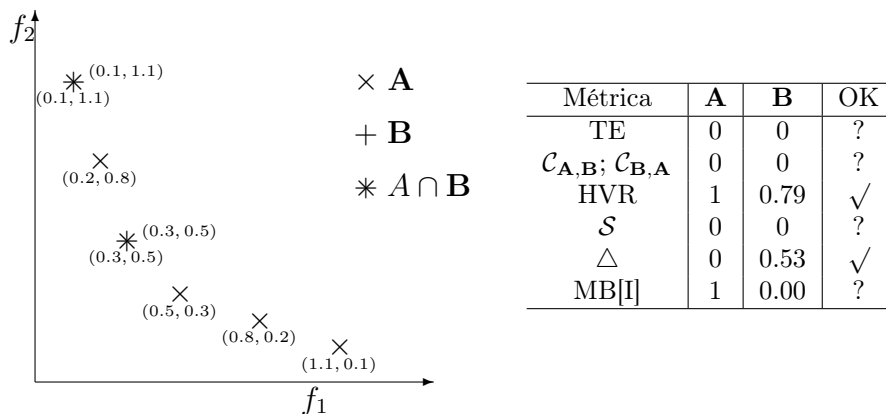
A Fig. 3.10 demonstra uma situação na qual os elementos de **A** dominam fortemente os elementos de **B**.

Neste caso, também não há dúvidas sobre a superioridade do conjunto **A**. As métricas convergem para mostrar esse resultado. Comparando com o Caso 1, observa-se que as métricas foram sensíveis à melhoria do conjunto **B** deste exemplo em relação ao anterior.

Fig. 3.10: **A** domina **B** fortemente.

### Caso 3: dominação fraca entre conjuntos solução

A Fig. 3.11 expõe um caso onde ambos conjuntos soluções retornam corretamente pontos na fronteira eficiente. No entanto, está claro que o resultado **A** é superior a **B**, pelo número de soluções geradas.

Fig. 3.11: **A** domina **B** fracamente.

As fronteiras eficientes **A** e **B** estão completamente sobre  $\mathbf{Y}^*$ , mas o conjunto **A** possui mais pontos que **B**. A métrica Espalhamento acusa distribuição de **B** inferior a de **A**, por **B** não possuir pontos nas extremidades. Caso os dois únicos pontos de **B** estivessem nos extremos, a métrica também retornaria  $\Delta_B = 0$  indicando igualdade com **A**, e então, somente a métrica Hipervolume acusaria a superioridade de **A**. MB[I] aponta que o conjunto

mais bem distribuído é **B**. No entanto, o número de soluções oferecidas em **A** deixa ele em vantagem em relação a **B**. Logo, a técnica MB[I] falha neste caso.

*Caso 4: conjuntos solução de desempenho equivalente*

A Fig. 3.12 exibe uma situação onde visualmente não há como afirmar que um grupo de soluções supera o outro. Pela figura, nenhuma fronteira é completamente eficiente.

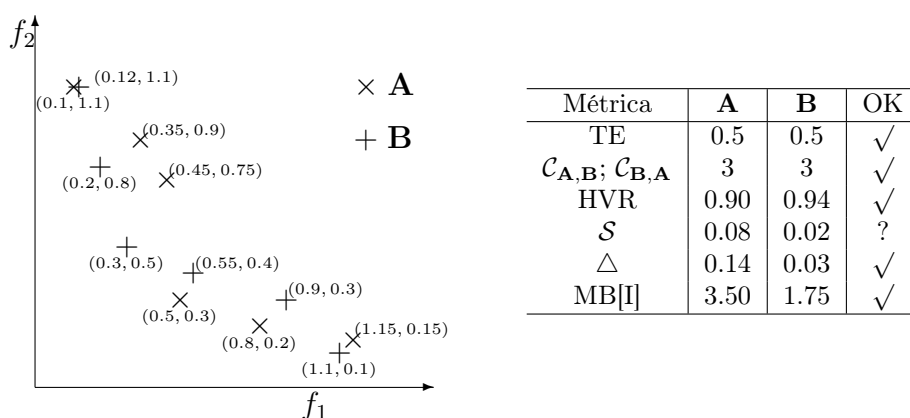


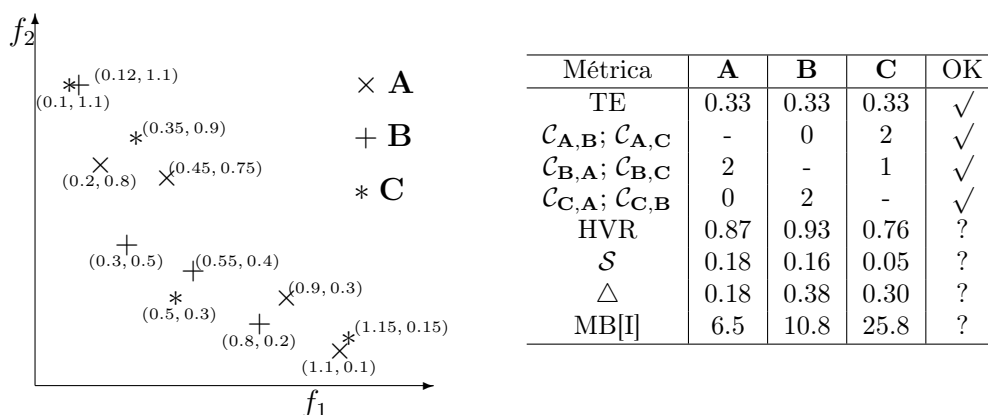
Fig. 3.12: **A** e **B** tem desempenho melhor sobre o outro em três pontos.

As métricas TE e  $\mathcal{C}$  concordam entre si apontando equivalência em desempenho do conjunto **A** em relação ao conjunto **B**. Por outro lado, as métricas que medem a distância entre as soluções apontam que o conjunto **B** contém soluções mais uniformemente distribuídas que **A**. A métrica do Hipervolume, destinada a medir tanto proximidade com  $\mathbf{Y}^*$  quanto distribuição, apresenta superioridade de **B**. Idem para MB[I].

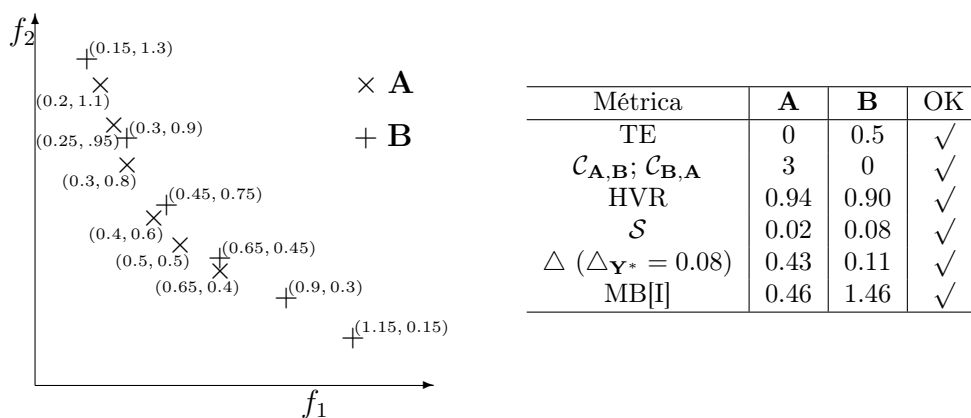
*Caso 5: comparação entre três conjuntos solução*

A Fig. 3.13 apresenta um caso interessante onde os conjuntos internamente não dominados **A**, **B** e **C** têm, aparentemente, desempenho superior um ao outro.

Ao comparar três conjuntos simultaneamente, as métricas TE e  $\mathcal{C}$  convergiram para indicar, novamente, equivalência entre os três conjuntos solução. Analisando as outras medições, a métrica  $\mathcal{S}$  apresenta o conjunto **C** como melhor,  $\Delta$  indica **A**, o HVR mostra **B** como melhor e MB[I] registra superioridade do conjunto **A**. Em outras palavras, não há convergência para o conjunto com a melhor distribuição.

Fig. 3.13: Os conjuntos **A**, **B** e **C** são, entre si, equivalentes.*Caso 6: uniformidade de distribuição nos conjuntos solução*

A Fig. 3.14 mostra uma situação onde **A** e **B** estão distribuídos uniformemente, porém **B** possui soluções em uma extensão superior a **A**.

Fig. 3.14: Uniformidade e extensibilidade das soluções de **A** e **B**.

Quatro métricas registram superioridade de **A**. Contudo, como foi abordado no Capítulo 4, é desejado que as soluções tenham soluções ao longo das extremidades da fronteira de Pareto, portanto seguindo este critério **B** tem conjunto mais representativo.

### *Conclusões sobre as métricas*

Nos casos de comparação de conjuntos apresentados pôde ser observado que as métricas atingem o objetivo ao qual foram propostas. No entanto, nenhuma delas cobre adequadamente e simultaneamente os quesitos de proximidade da fronteira de Pareto e de distribuição uniforme dentro do conjunto solução. Dentre as métricas apresentadas, para o capítulo de testes decidiu-se por empregar a métrica TE para medir proximidade e MB[I] para medir uniformidade de conjuntos solução pela simplicidade de implementação e pelos poucos parâmetros exigidos.

### *3.8 Considerações Finais*

Conforme discutido neste capítulo, são várias as formas de se abordar a otimização robusta. A forma adotada nesta Tese, descreve a otimização robusta como um problema min max, ou seja, como a minimização da máxima interferência da incerteza. Em outras palavras, o problema de otimização robusta consiste em encontrar os minimizadores com os menores valores de desempenho quando observado pior caso de incerteza. Uso de Análise Intervalar para contabilização da ação das incertezas é vantajoso quando o parâmetro de incertezas aparece apenas uma vez na expressão formal do problema de otimização. Nestes casos a função objetivo de inclusão é mínima e a computação das incertezas é feita com apenas um cálculo. Por outro lado, quando há múltiplas ocorrências do parâmetro de incerteza, a função objetivo de inclusão mínima é gerada a partir da divisão do parâmetro de incertezas em subpavimentos e da computação isolada de cada um destes subpavimentos. A precisão da função de inclusão e o esforço computacional crescem com o número de subpavimentos, podendo deixar a aplicação do método intervalar inviável.

Três novos métodos de otimização foram apresentados para resolver o problema robusto proposto, são eles: a) o [I]RMOA I; o [I]RMOA II; e o [I]RMOEA. Todos empregam técnicas intervalares para computar a incerteza e para realizar outros processos internos, no entanto, cada um tem suas particularidades. As considerações sobre cada um deles é dada a seguir.

O [I]RMOA I foi desenvolvido para definir um envelope sobre os minimizadores robustos ao invés de encontrá-los. O envelope é construído da seguinte forma: a partir da amostragem do espaço de busca e do espaço de incertezas, o [I]RMOA I classifica cada exemplar como pertencente às regiões superior ou inferior à fronteira robusta. Observa-se que classificar um exemplar como pertencente à região inferior à fronteira robusta conduz a um custo computacional elevado, pois os espaços de busca e de incertezas

devem ser continuamente bisseccionados e investigados. Nota-se que espaços multi-dimensionais sejam eles de busca e/ou de incertezas deixam o processo ainda mais caro. Dentre os pontos fortes do método, destacam-se a simplicidade, o uso de poucos parâmetros e o fornecimento de uma região onde os pontos da fronteira de Pareto robusta se encontram, segundo dada precisão do método. Por causa deste último item, o [I]RMOA I pode ser interpretado como um método de validação.

O [I]RMOA II computa intervalos diretamente nas funções objetivo durante o processo de busca dos minimizadores robustos. Em poucas palavras, o algoritmo subdivide os espaços de busca e de incertezas, utiliza estes intervalos como argumentos das funções objetivo de inclusão e realiza as comparações de dominância usando os intervalos gerados pelas funções objetivo de inclusão. Empregando uma filosofia “pessimista”, somente são excluídos os intervalos que, seguramente, não contêm minimizadores. Como consequência, o [I]RMOA II garante encontrar todos os minimizadores robustos. A garantia do método depende das funções objetivo de inclusão; quando for possível assumir que elas sejam monotônicas e convergentes (veja a Fig. 2.5) o [I]RMOA II converge garantidamente para o conjunto solução. Mas, a garantia do [I]RMOA II custa caro; qualquer intervalo com uma ínfima probabilidade de conter minimizadores robustos não pode ser excluído. Assim o [I]RMOA II mantém armazenado em sua estrutura de dados todas as caixas intervalares candidatas a conterem minimizadores. Se fosse utilizada uma filosofia “otimista” (2.8) a estrutura de dados poderia ser otimizada, o esforço computacional seria menor, mas em contrapartida, a garantia de encontrar todos os minimizadores seria perdida.

O [I]RMOEA é um método híbrido que associa intervalos a algoritmos evolucionários, especificamente, o grupo dos Algoritmos Genéticos. O processo de busca das soluções robustas é composto por etapas tradicionais dessa classe de algoritmos, sendo que os vínculos com os métodos intervalares ocorrem na computação da incerteza, no tratamento das funções de restrição, no processo de seleção e na escolha de soluções na fronteira robusta. No último caso, quando a fronteira robusta tem mais soluções candidatas que o número máximo estabelecido por  $n_{pop}$ . O custo computacional é fixo por geração e menos sensível à dimensionalidade dos espaços de busca e de incertezas que os demais algoritmos apresentados.

Além dos algoritmos, este capítulo apresentou como pode ser feito o tratamento de restrições com métodos intervalares. As funções de restrição são escritas na forma de funções de inclusão. Assumindo-se que estas funções são monotônicas e convergentes, elas asseguram a exclusão de porções não viáveis do espaço de busca. Claramente, elas são também dependentes da dimensionalidade dos espaços de busca e de incertezas.

As técnicas de nicho tradicionais buscam descobrir informações de proximidade entre as soluções. O esforço para medir essas distâncias é considerado como “alto” por diversos pesquisadores. Refletindo sobre essa questão de custo, elaborou-se a técnica de Nicho por Bissecção Intervalar NB[I] que usa um mecanismo diferente para criar nicho que é independente da medida da distância entre as soluções. As bissecções no espaço dos objetivos separam a cada iteração os nichos em duas partes. As soluções internas em cada nicho são também separadas nos novos nichos gerados de acordo com os seus valores de desempenho. Sobre as soluções que participam de um mesmo nicho, independente critério adotado, pode-se afirmar que elas são vizinhas segundo aquele critério. Em princípio, os nichos deixam pontos vizinhos agrupados, mas há exceções como pôde ser visto na Fig. 3.7. O único parâmetro é o número de rodadas de bissecção  $r_{bis}$ , que na verdade é o critério de parada. Outros critérios de parada podem substituir  $r_{bis}$ , como por exemplo a definição de limites inferior e superior para o volume dos nichos.

A métrica por bissecção de intervalos MB[I] foi desenvolvida para medir a uniformidade de conjuntos compostos por elementos não dominados entre si. Sucintamente, a métrica bissecciona o espaço dos objetivos até que todas as soluções estejam “unidas” por uma caixa intervalar. A uniformidade do conjunto é medida pelo desvio padrão dos volumes dessas caixas. No entanto, o resultado deve ser comparado com um valor de referência calculado sobre uma fronteira de Pareto cujos pontos sejam conhecidos, ou ainda, a métrica pode ser utilizada para comparar dois conjuntos soluções quaisquer. Menor valor de desvio padrão, melhor distribuição de conjunto solução.

## 4. VALIDAÇÃO NUMÉRICA DOS ALGORITMOS PARA OTIMIZAÇÃO ROBUSTOS

A otimização robusta tem uma formulação mais complexa que a otimização tradicional, e conseqüentemente, os métodos para resolvê-la são também mais elaborados. Os algoritmos para tratar problemas de otimização robusta apresentados neste trabalho carecem agora de serem validados. Neste contexto, o termo “validação numérica” é empregado para representar os procedimentos de observação, comparação e avaliação do grau de fidelidade dos algoritmos com suas especificações técnicas, através de simulações implementadas com uso de funções teste de minimização. Os métodos são considerados “validados” se for comprovado que eles conseguem resolver, dentro dos limites de precisão definidos, os problemas teste propostos. Além da validação, foram analisados outros aspectos considerados importantes para avaliação de quaisquer métodos numéricos como a flexibilidade de adaptação a problemas diferentes, simplicidade de escolha dos parâmetros e esforço computacional para executar a tarefa de otimização.

Geralmente, a validação dos algoritmos é feita comparando-se com outros similares anteriormente criados para resolver o mesmo tipo de problema. Entretanto, não foram encontrados métodos intervalares para o modelo de otimização multi-objetivo robusta definido nas expressões (3.9-3.18). Outros métodos, como descrito por Kouvelis e Yu [75], resolvem a tarefas de maximização da incerteza e minimização das funções objetivo utilizando-se métodos puramente determinísticos que analisam todo o espaço de busca. Adicionalmente, se as funções objetivo e de restrições do problema robusto formarem uma região convexa, certamente algoritmos quasi-Newton ou que usam direções conjugadas do gradiente e planos de corte podem resolver o problema de forma confiável e eficiente. Boyd e Vandenberghe [13] (pg.13), apresentaram os métodos de pontos interiores como as melhores opções para otimização convexa. Importante observar, que os métodos propostos aqui não competem em velocidade e precisão com os métodos de otimização convexa pois é inversamente proporcional a relação entre esses dois quesitos nos métodos intervalares. Os pontos fortes dos métodos intervalares são: a) independência da convexidade e da continuidade da fronteira Pareto robusta; b) independência do cálculo de derivadas nas funções objetivo; c) globalidade

da busca; d) garantia de computação do pior caso da incerteza; e) garantia de envelopar a fronteira robusta nos métodos [I]RMOA I e [I]RMOA II; f) baixa quantidade de parâmetros; e g) simplicidade dos algoritmos. Por serem grandes as diferenças entre os algoritmos e formulações robustas existentes, decidiu-se por elaborar testes independentes de estudos anteriores.

A natureza dos problemas teste implementados cobrem diversos aspectos de complexidades citados acima. A próxima seção descreve os detalhes de como foram organizados os testes e apresentação dos resultados.

### 4.1 Metodologia

Naturalmente, o desempenho dos algoritmos é sensível à estrutura dos problemas teste. Assim, foram escolhidas 5 funções com diferentes questões de complexidade. Considerando o espaço dos objetivos, esse conjunto de funções cobrem situações de multimodalidade no espaço dos objetivos, onde as fronteiras de Pareto robusta podem ter forma convexa ou não convexa, contínua ou descontínua, e algumas possíveis combinações destas características.

Os testes foram organizados e especificados de tal forma que seja facilitada a compreensão do funcionamento dos algoritmos. Os problemas multi-objetivos escolhidos para teste se limitam a casos com dois critérios para auxiliar a análise visual do espaço dos objetivos. As funções teste foram expressadas formalmente sendo descritas algumas de suas características de complexidade com seus respectivos conjuntos solução. Observa-se que as funções utilizadas originaram-se de problemas de otimização tradicional, sendo os parâmetros de incerteza inseridos posteriormente. O efeito da incerteza nas funções teste também foi interpretado. No próximo passo, são definidos os valores dos parâmetros fixos de cada algoritmo, para cada problema. Finalmente, os experimentos são realizados, os resultados são mostrados em gráficos, e seguida discutidos.

### 4.2 Funções Teste

Como discutido no texto introdutório do Capítulo 3, a definição do problema de otimização robusta depende da interpretação do projetista. Logo, as funções teste para otimização robusta devem estar em sintonia com os mesmos conceitos da definição formal. Como consequência disso, decidiu-se por adaptar funções teste endereçadas para algoritmos multi-objetivos tradicionais uma vez que não foram encontradas funções teste para otimização robusta que sejam referência para o modelo de otimização utilizado neste trabalho.

Além da finalidade de encontrar soluções Pareto, os métodos de otimização multi-objetivo tradicionais também têm sido projetados para maximizar o número de pontos não dominados e ainda que estes pontos estejam espaçados o mais uniformemente possível. Funções teste que não tragam consigo nenhum empecilho quanto a estes dois fatores, podem ser consideradas não eficazes para comparação de desempenho entre algoritmos. Por exemplo, quando existe função objetivo multimodal ou com pontos Pareto isolados, os algoritmos terão mais dificuldade para ter sucesso na busca pela fronteira repleta de pontos homoganeamente distribuídos. Da mesma forma, descontinuidade e não convexidade da fronteira Pareto, presença de funções de restrição e dispersão das soluções no espaço de busca, são fatores que podem complicar a representatividade das soluções eficientes ao longo da fronteira eficiente. Além disso, o desempenho dos métodos intervalares fica comprometido quando a função objetivo de inclusão não for estreita. Em outras palavras, as ocorrências múltiplas de mesmos parâmetros na função objetivo aumenta o pessimismo na computação da função a qual produz um intervalo mais “largo” que é propagado ao longo do processo de otimização. Como consequência, os métodos intervalares perdem em precisão e desempenho. Os casos teste apresentados aqui simulam algumas situações citadas acima onde os algoritmos podem encontrar dificuldade de realizar a busca pelos pontos ótimos. As funções teste desta seção foram utilizadas como referência em várias publicações, como por exemplo Barichard e Hao [6], Coello e Sierra [21] e Deb *et al.* [34]. Para encontrar mais detalhes sobre funções teste, veja Deb [32]. As funções teste estão descritas abaixo em duas formas: a) versão tradicional; e b) versão adaptada para otimização robusta.

**Função Schaffer 2 (SCH2)** - A função Schaffer 2 [103] é irrestrita, utiliza uma única variável e se caracteriza pela descontinuidade da fronteira de Pareto. A forma tradicional da função é definida pelas expressões

$$\min_{x \in [-5, 10]} f_1(x) = \begin{cases} -x, & \text{se } x \leq 1 \\ x - 2, & \text{se } 1 < x \leq 3 \\ 4 - x, & \text{se } 3 < x \leq 4 \\ x - 4, & \text{se } x > 4; \end{cases} \quad (4.1)$$

$$f_2(x) = (x - 5)^2.$$

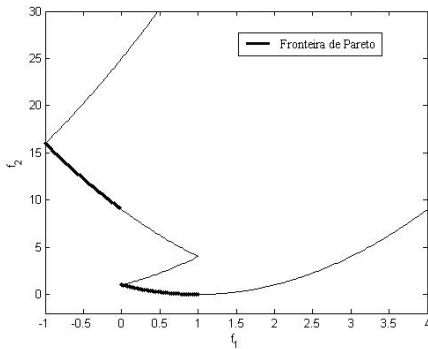
A fronteira de Pareto está destacada na Fig. 4.1(a) e ocorre quando  $\mathbf{X}^* \in \{[1, 2] \cup [4, 5]\}$ . A versão robusta de SCH2, veja Fig. 4.1(b), foi criada

adicionando-se o parâmetro de incerteza  $\mathbf{p}$  conforme mostrado abaixo

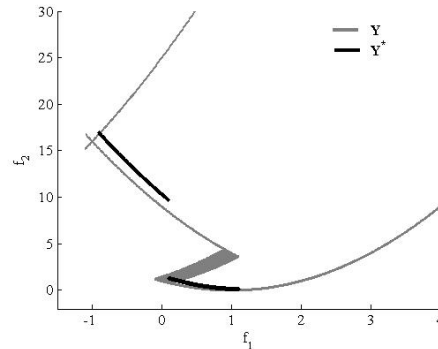
$$\min_{x \in [-5, 10]} \max_{\mathbf{p} \in [-0.1, 0.1]} f_1(x, \mathbf{p}) = \begin{cases} -(x + p_1), & \text{se } x \leq 1 \\ (x + p_1) - 2, & \text{se } 1 < x \leq 3 \\ 4 - (x + p_1), & \text{se } 3 < x \leq 4 \\ (x + p_1) - 4, & \text{se } x > 4; \end{cases} \quad (4.2)$$

$$f_2(x, \mathbf{p}) = ((x + p_1) - 5)^2 + p_2.$$

O parâmetro  $p_1$  pode ser interpretado como incerteza paramétrica e  $p_2$  como uma imprecisão na medida de  $f_2$ . O conjunto solução permanece inalterado  $\mathbf{X}^* \in \{[1, 2] \cup [4, 5]\}$ , mesmo com a presença de incertezas. Importante observar que para dada solução  $x^*$ , diferentes  $\mathbf{p}$  podem produzir opções ótimas não dominadas entre si. Por exemplo, considere a solução robusta  $x_1^* = 4.34$ . O pior caso para  $f_1(x_1^*, \mathbf{p})$  ocorre quando  $p_1 = 0.1$  e para  $f_2(x_1^*, \mathbf{p})$  quando  $\mathbf{p} = \{-0.1, 0.1\}$ . Note que o pior caso de  $f_1$  e de  $f_2$  não podem acontecer ao mesmo tempo, pois usam diferentes instâncias de  $p_1$ . Como resultado, a fronteira robusta  $\mathbf{Y}^* \not\subset \mathbf{f}(x, \mathbf{p})$ . Observa-se ainda que a versão robusta da função SCH contém apenas simples ocorrências de  $x$  e de  $\mathbf{p}$ . Logo a função de inclusão neste caso é considerada estreita.



(a) SCH2 - versão original [103].



(b) SCH2 - versão robusta.

Fig. 4.1: Funções SCH2. Em ambas figuras, a cor cinza representa o espaço dos objetivos; a cor preta as fronteiras de Pareto (a) e fronteira de Pareto robusta (b). Observe que somente alguns pontos da fronteira robusta pertencem a imagem de  $\mathbf{f}(x, \mathbf{p})$ .

**Função Fonseca e Fleming (FON)** - Fonseca e Fleming [43] apresentaram uma função de teste irrestrita, exponencial, não convexa de  $n$  variáveis e dois

objetivos descrita por

$$\begin{aligned} \min_{\mathbf{x} \in [-4,4]} \quad & f_1(\mathbf{x}) = 1 - e^{-\sum_{i=1}^n (x_i - \frac{1}{\sqrt{n}})^2}; \\ & f_2(\mathbf{x}) = 1 - e^{-\sum_{i=1}^n (x_i + \frac{1}{\sqrt{n}})^2}. \end{aligned} \tag{4.3}$$

Pelas expressões acima nota-se que os menores valores de desempenho em  $f_1$  e  $f_2$  ocorrem quando  $x_i = 1/\sqrt{n}$  e  $x_i = -1/\sqrt{n}$ , respectivamente. O espaço dos objetivos da função FON é mostrado na Fig. 4.2(a). A imagem da função foi plotada amostrando-se o espaço de busca e de incerteza uniformemente. No entanto, o gráfico apresenta uma maior concentração de pontos nas regiões mais afastadas da fronteira de Pareto. Esta é uma consequência da contribuição exponencial na função objetivo e ainda é um fator que dificulta a convergência para a fronteira de Pareto. A versão robusta da função

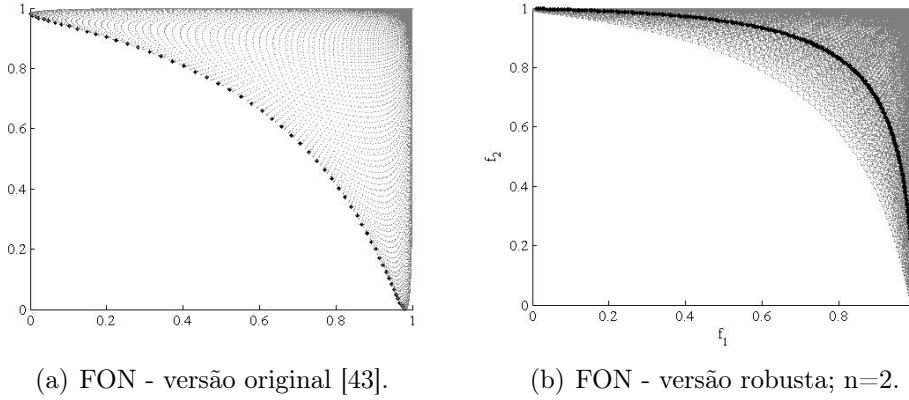


Fig. 4.2: Funções FON. Em ambas figuras, a cor cinza representa o espaço dos objetivos; a cor preta as fronteiras de Pareto (a) e fronteira de Pareto robusta (b). Entretanto, neste caso é fácil perceber que  $\mathbf{Y}^* \subset \mathbf{f}(\mathbf{x}, p)$ .

FON foi construída assumindo-se a incerteza conforme mostrado abaixo

$$\begin{aligned} \min_{\mathbf{x} \in [-4,4]} \max_{p \in [1.1, 1.3]} \quad & f_1(\mathbf{x}, p) = 1 - e^{-\sum_{i=1}^n (x_i - \frac{p}{\sqrt{n}})^2}; \\ & f_2(\mathbf{x}, p) = 1 - e^{-\sum_{i=1}^n (x_i + \frac{p}{\sqrt{n}})^2}. \end{aligned} \tag{4.4}$$

Similarmente, os menores valores de desempenho ocorrem em  $f_1$  e  $f_2$  quando  $x_i = p/\sqrt{n}$  e  $x_i = -p/\sqrt{n}$ , respectivamente. Mas o pior caso de cada uma dessas funções, para dada solução  $x_i$ , acontece sempre no maior valor de  $p$ . Neste problema,  $p$  pode ser considerado como uma incerteza paramétrica. Nota-se em (4.4) que o parâmetro  $\mathbf{p}$  é computado independentemente para cada  $x_i$ . Logo a função de inclusão neste caso é considerada estreita.

**Família de funções ZDT** - Deb [31] apresentou um trabalho sobre construção de funções teste 2D. O gerador de funções é descrito pelas equações a seguir

$$\begin{aligned} \min_{\mathbf{x} \in \mathbf{X}} \quad & f_1(\mathbf{x}) = f_1(x_1, x_2, \dots, x_k); \\ & f_2(\mathbf{x}) = h'(x_{k+1}, \dots, x_n) \times h(f_1, h'). \end{aligned} \tag{4.5}$$

Manipulando adequadamente (4.5), criam-se funções teste com as complicações citadas anteriormente. Por exemplo, dependendo do valor de  $h$  as características de não convexidade e descontinuidade são introduzidas na fronteira Pareto. Escolhendo  $h'$  apropriadamente, tal como uma função multimodal, dificulta-se a convergência para soluções não dominadas. Por outro lado, se  $f_1$  for não linear ou multi-dimensional, as soluções estarão mais dispersas. Através da (4.5) Zitzler, Deb e Thiele criaram em [132] a família de funções ZDT. Foram escolhidas três delas para realização dos testes. A primeira, ZDT1 é descrita como

$$\begin{aligned} f_1(\mathbf{x}) &= x_1, \\ h'(\mathbf{x}) &= 1 + \frac{9}{n-1} \sum_{i=2}^n x_i, \\ h(f_1, h') &= \sqrt{f_1/h'}. \end{aligned} \tag{4.6}$$

sendo  $n = 1, \dots, 30$  e o conjunto solução é formado por  $x_1 \in [0, 1]$  e  $x_i = 0$ ,  $i = 2, \dots, n$ . Como pode ser visto na Fig. 4.3, a região não dominada no espaço dos objetivos é convexa.

O parâmetro de incerteza foi introduzido em dois pontos do problema ZDT1: a) como incerteza paramétrica em  $f_1$ ; e b) como imprecisão na formulação de  $f_2$ . As expressões em (4.7) apresentam o caso robusto para ZDT1.

$$\begin{aligned} \min_{\mathbf{x} \in [0,1]} \quad & \max_{\mathbf{p} \in [0,0.05]} \quad f_1(\mathbf{x}, \mathbf{p}) = x_1 + p_1; \\ & f_2(\mathbf{x}, \mathbf{p}) = h'(\mathbf{x}) \times h(f_1, h', \mathbf{p}). \end{aligned} \tag{4.7}$$

$$\begin{aligned} \text{sendo :} \quad & h'(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i, \\ & h(f_1, h', \mathbf{p}) = 1 - \sqrt{f_1/h'} + p_2. \end{aligned}$$

Observa-se em (4.7) que os parâmetros de busca e incerteza não são computados em multiplicidade. Assim, também para esse caso a função de inclusão pode ser considerada estreita.

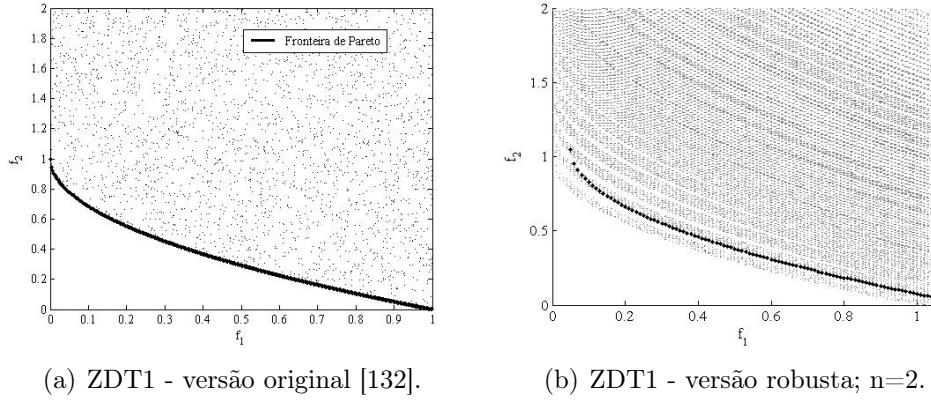


Fig. 4.3: Funções ZDT1. Em (a) a fronteira de Pareto estende-se no intervalo de  $[0, 1]$  para  $f_1$  e  $f_2$ . Em (b), fica claro que a fronteira robusta sofre um deslocamento para direita em função do efeito aditivo do parâmetro de incerteza.

A função ZDT2, descrita abaixo, foi construída similarmente à ZDT1, diferenciando-se pela expressão formal da função  $h$ . No entanto, a mudança de  $h$  gerou uma fronteira de Pareto com forma não convexa, como pode ser observado na Fig. 4.4.

$$\begin{aligned}
 f_1(\mathbf{x}) &= x_1, \\
 h'(\mathbf{x}) &= 1 + \frac{9}{n-1} \sum_{i=2}^n x_i, \\
 h(f_1, h') &= 1 - (f_1/h')^2.
 \end{aligned} \tag{4.8}$$

Novamente,  $n = 1, \dots, 30$  e o conjunto solução é formado por  $x_1 \in [0, 1]$  e  $x_i = 0, i = 2, \dots, n$ . A versão robusta da função ZDT2 é mostrada na Fig. 4.4(b) e descrita formalmente em (4.9).

$$\begin{aligned}
 \min_{\mathbf{x} \in [0,1]} \max_{\mathbf{p} \in [-0.05, 0.05]} \quad & f_1(\mathbf{x}, p) = x_1; \\
 & f_2(\mathbf{x}, p) = h'(\mathbf{x}, p) \times h(f_1, h', p). \\
 \text{s.a.} \quad & g(\mathbf{x}) = \sqrt{(x_1 - 0.7)^2 + x_2^2 + \dots + x_n^2} > 0.1
 \end{aligned} \tag{4.9}$$

$$\begin{aligned}
 \text{sendo :} \quad & h'(\mathbf{x}, p) = 1 + \frac{9}{n-1} (\sum_{i=2}^n x_i) + p, \\
 & h(f_1, h', p) = 1 - (f_1/h')^2 + p.
 \end{aligned}$$

Neste caso,  $p$  representa imprecisão na computação de partes distintas da função  $f_2$ . Note que, para dado  $\mathbf{x}$ ,  $h'$  cresce com crescimento de  $p$  e como

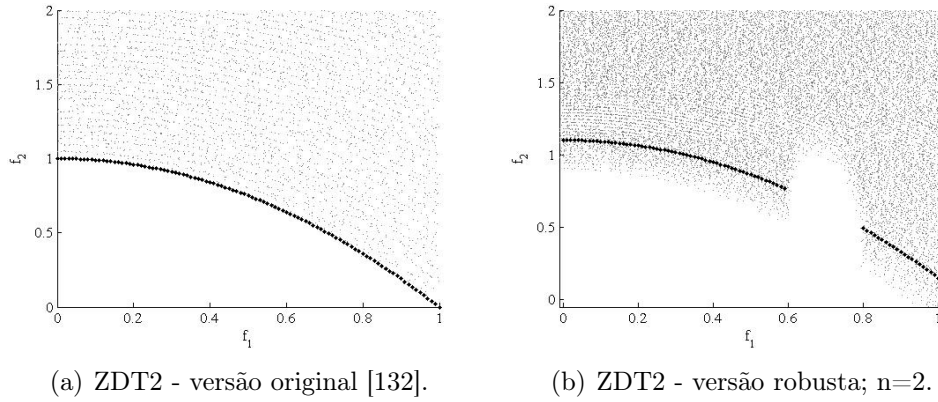


Fig. 4.4: Funções ZDT2. Observa-se em (b) a ação da função de restrição; parte do espaço dos objetivos e da fronteira de Pareto robusta são excluídos por serem não viáveis.

consequência  $h$  decresce. Por outro lado, a segunda contribuição de  $p$  produz um efeito direto a  $f_2$ ;  $f_2$  tem maior valor de desempenho quando o valor de incerteza  $p$  é maior. Logo,  $p$  age de forma contraditória dentro de  $f_2$ . Além do parâmetro de incerteza, foi introduzido a função de restrição quadrática  $g(\mathbf{x})$ . Observe que a restrição deixa inviável uma parte de fronteira robusta.

Na expressão robusta da função ZDT2 (4.9) verifica-se a multiplicidade do parâmetro  $p$ . Logo, a função de inclusão não pode ser considerada estreita.

A função ZDT3, mostrada na Fig. 4.5, tem como características a multimodalidade no espaço dos objetivos e a fronteira de Pareto descontínua. ZDT3 é dada por

$$\begin{aligned}
 f_1(\mathbf{x}) &= x_1, \\
 h'(\mathbf{x}) &= 1 + \frac{9}{n-1} \sum_{i=2}^n x_i, \\
 h(f_1, h') &= 1 - \sqrt{f_1/h'} - (f_1/h') \text{sen}(10\pi f_1).
 \end{aligned} \tag{4.10}$$

Como em ZDT1 e ZDT2,  $n = 1, \dots, 30$  e a fronteira de Pareto ocorre quando  $x_1 \in [0, 1]$  e  $x_2, x_3, \dots, x_n = 0$ .

A formulação robusta foi obtida inserindo-se o parâmetro de incerteza na função  $f_2$  conforme mostrado na formulação 4.11. Observe que também

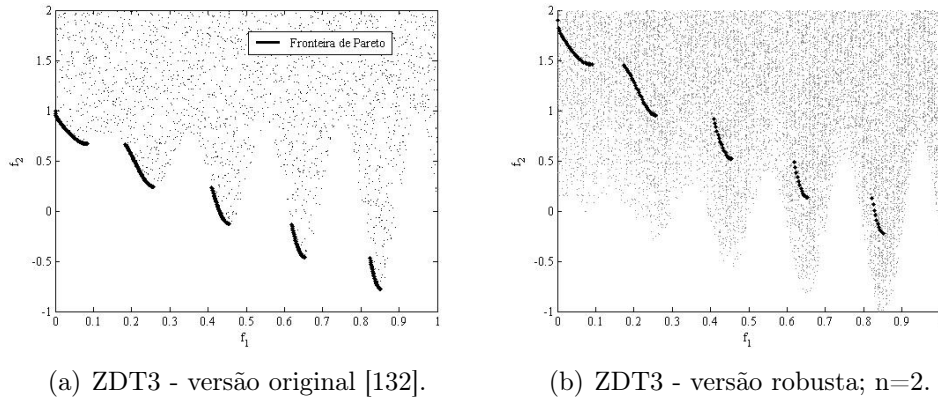


Fig. 4.5: Funções ZDT3. Em (b), observe que a descontinuidade na fronteira de Pareto robusta é mantida na versão robusta.

neste caso,  $p$  representa imprecisão na computação de  $f_2$ .

$$\begin{aligned} \min_{\mathbf{x} \in [0,1]} \max_{\mathbf{p} \in [-0.1,0.1]} f_1(\mathbf{x},) &= x_1; \\ f_2(\mathbf{x}) &= h'(\mathbf{x}, p) \times h(f_1, h', p). \end{aligned} \tag{4.11}$$

$$\begin{aligned} \text{sendo :} \quad h'(\mathbf{x}, p) &= 1 + \frac{9}{n-1} \left( \left( \sum_{i=2}^n x_i \right) + p \right), \\ h(f_1, h', p) &= 1 - \sqrt{f_1/h'} - (f_1/h') \text{sen}(10\pi f_1). \end{aligned}$$

A função ZDT3 (4.11) contém repetidas ocorrências tanto de  $x_1$  quanto de  $p$ . Logo, a função de inclusão neste caso não pode ser considerada estreita.

A partir da apresentação das funções teste, a próxima etapa descreve os experimentos com os algoritmos robustos e os resultados obtidos com cada função teste.

### 4.3 Testes e Resultados

Em um primeiro momento, os testes foram voltados ao estudo comparativo dos métodos, onde os mesmos casos de teste foram realizados. A finalidade foi acompanhar o comportamento de cada algoritmo frente à variação de alguns de seus parâmetros. Na seqüência, alguns testes independentes para cada algoritmo são executados, sendo que o foco principal está nas questões de desempenho e complexidade computacional.

Os critérios para a avaliação dos métodos foram definidos como se segue. Primeiramente, determinou-se uma aproximação para o conjunto de mini-

mizadores robustos  $\mathbf{X}^*$ , o conjunto solução discreto  $\mathbf{X}_d^* \subset \mathbf{X}^*$ , para servir de comparação. Alternativamente, utilizou-se a imagem de  $\mathbf{X}_d^*$ ,  $\mathbf{Y}_d^* \subset \mathbf{Y}^*$ , quando a discussão dos resultados foi realizada no espaço dos objetivos. Os conjuntos  $\mathbf{X}_d^*$  e  $\mathbf{Y}_d^*$  de cada função teste foram obtidos com a discretização dos espaços de busca e de incerteza, seguida pela computação das funções objetivo e das fronteiras robustas. Finalizando as notações, o conjunto solução encontrado pelos algoritmos e sua respectiva imagem foram simbolizados por  $\mathbf{X}_s^*$  e  $\mathbf{Y}_s^*$ . Em segundo lugar, as avaliações dos métodos foram baseadas nas propostas de cada algoritmo. A proposta do [I]RMOA I é de envolver a fronteira de Pareto robusta por duas regiões demarcadas pelos elementos de  $\mathbf{K}^-$  e  $\mathbf{K}^+$ , segundo dada precisão  $\varepsilon_{[\mathbf{x}]}$ . Então, o [I]RMOA I será considerado aprovado quando  $\mathbf{Y}_d^* \subset [\mathbf{Y}'] / (\mathbf{K}^- \cup \mathbf{K}^+)$ . Por outro lado, os algoritmos [I]RMOA II e [I]RMOEA têm a finalidade de encontrar o conjunto solução para a otimização robusta. Para esses casos utilizou-se, quando necessário, a métrica da TE para quantificar a proximidade da fronteira robusta encontrada com  $\mathbf{Y}_s^*$  e a métrica por bissecção intervalar MB[I] para qualificar a distribuição dos pontos na fronteira. No entanto, nos testes do [I]RMOA II, a análise de proximidade do conjunto solução foi feita por meio de gráficos, pois as informações contidas neles foram suficientes para concluir sobre a eficiência do método. A Tab.4.1 resume as características dos problemas robustos testados e em seguida, os testes são iniciados.

Tab. 4.1: Sumário dos problemas testes para otimização robusta.

Função	$\mathbf{f}(\mathbf{x}, \mathbf{p})$ , $\mathbf{g}(\mathbf{x}, \mathbf{p})$	$[\mathbf{X}] \in \mathbb{IR}^{n_x}$	$[\mathbf{P}] \in \mathbb{IR}^{n_p}$	$n_x$	$n_p$
SCH2	(4.2)	$[-5, 10]$	$[-\mathbf{0.1}, \mathbf{0.1}]$	1	2
FON	(4.4)	$[-\mathbf{4}, \mathbf{4}]$	$[1.1, 1.3]$	2	1
ZDT1	(4.5) e (4.7)	$[\mathbf{0}, \mathbf{1}]$	$[\mathbf{0}, \mathbf{0.05}]$	2	2
ZDT2	(4.5) e (4.9)	$[\mathbf{0}, \mathbf{1}]$	$[-0.05, 0.05]$	2	1
ZDT3	(4.5) e (4.11)	$[\mathbf{0}, \mathbf{1}]$	$[-0.1, 0.1]$	2	1

### 4.3.1 [I]RMOA I

Para cada função teste três tipos de resultados estão apresentados: a) a quantidade de elementos dos conjuntos  $\mathbf{K}^-$  e  $\mathbf{K}^+$  que demarcam o envelope sobre a fronteira robusta segundo o número de pontos da discretização do espaço de busca; b) o esforço computacional tanto para discretização do espaço de busca, quanto nos processos mais internos do [I]RMOA I; e c) qualidade do envelope sobre  $\mathbf{Y}_d^*$  segundo o número de pontos em  $\mathbf{K}^-$  e  $\mathbf{K}^+$ . Em outras palavras, nestes experimentos foi examinado como o [I]RMOA I se comporta diante dos parâmetros  $\varepsilon_{\mathbf{x}\#}$  e  $\varepsilon_{[\mathbf{x}]}$ .

[I]RMOA I  $\times$  SCH2

Nas simulações, empregaram-se  $\varepsilon_{x\#} = [0.02 : 0.02 : 0.1]$  e  $\varepsilon_{[x]} = [0.02 : 0.02 : 0.1]$ . A Fig. 4.6 mostra alguns exemplos de como o número de elementos em  $\mathbf{K}^-$  e  $\mathbf{K}^+$  variam com  $\varepsilon_{x\#}$  e  $\varepsilon_{[x]}$ .

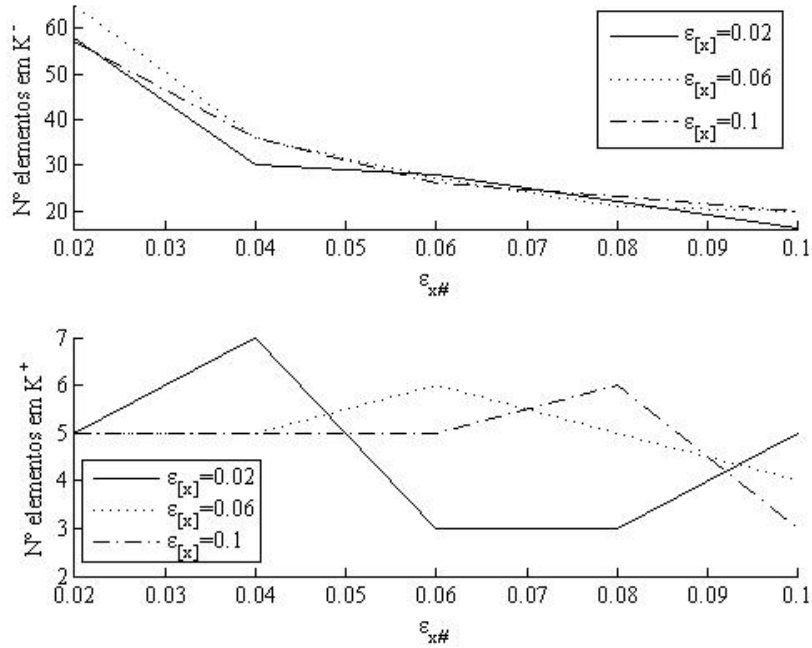


Fig. 4.6: SCH2 - [I]RMOA I:  $\mathbf{K}^-$  e  $\mathbf{K}^+ \times \varepsilon_{x\#}$ . Acima, o número de pontos no conjunto  $\mathbf{K}^-$  diminui com o número de amostras. Abaixo, a quantidade de pontos de  $\mathbf{K}^+$  não apresenta tendência com a variação de  $\varepsilon_{x\#}$ .

Pela Fig. 4.1(b) nota-se que não são necessários muitos pontos em  $\mathbf{K}^+$  para definir  $\mathbf{Y}^+$ . Assim, a relação entre  $\mathbf{K}^+$  e  $\varepsilon_{x\#}$  não apresenta nenhuma tendência. Por outro lado, o conjunto  $\mathbf{K}^-$  apresentou-se sensível ao número de pontos da função grade.

O custo computacional no [I]RMOA I foi considerado elevado, veja a Fig. 4.7, seja para a realização da discretização inicial com  $\mathbf{f}(x, \mathbf{p})$ , seja no processo de busca por soluções com  $[\mathbf{f}](c([x]), [\mathbf{p}])$  e  $[\mathbf{f}](c([x]), \mathbf{c}([\mathbf{p}]))$ .

A qualidade dos envelopes sobre  $\mathbf{Y}_d^*$  pode ser observada na Fig. 4.8. De acordo com as informações nas figuras Fig. 4.6 e Fig. 4.8 não percebe-se claramente que o envelope sobre a fronteira robusta é mais preciso para as simulações com maior quantidade de elementos em  $\mathbf{K}^-$  e  $\mathbf{K}^+$ .

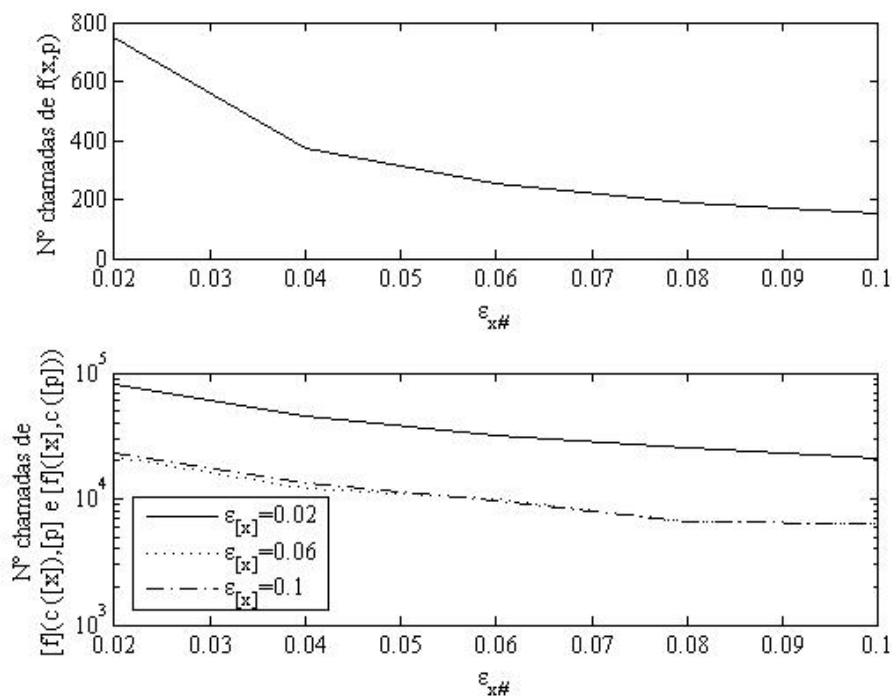


Fig. 4.7: SCH2 - [I]RMOA I: custo computacional. Acima, o esforço na função grade em número de cálculos de  $\mathbf{f}(x, \mathbf{p})$ . Abaixo, o número de chamadas das funções  $[\mathbf{f}](c(x), \mathbf{p})$  e  $[\mathbf{f}](x, \mathbf{c}(\mathbf{p}))$  (Tab.3.4, linhas 3 e 4).

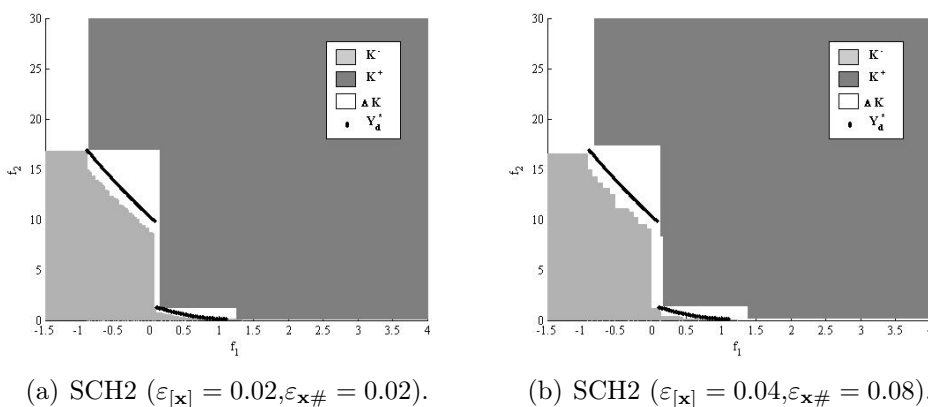


Fig. 4.8: SCH2 - [I]RMOA I: envelopes sobre  $\mathbf{Y}_d^*$ . Por se tratar de uma fronteira simples para ser envelopada, não está visualmente claro que o aumento de precisão resulta em melhoria de qualidade no envelope.

[I]RMOA I  $\times$  FON

Para a função FON, foram utilizados  $\varepsilon_{x\#} = [0.05, 0.1, 0.2 : 0.2 : 1]$  e  $\varepsilon_{[x]} = [0.2 : 0.2 : 1]$ . Os resultados encontram-se nas Fig. 4.9, Fig. 4.10 e Fig. 4.11.

Examinando a Fig. 4.9 constata-se que o número de elementos de  $\mathbf{K}^-$  e  $\mathbf{K}^+$  decresce fortemente nos primeiros pontos no gráfico. Para evidenciar este comportamento, decidiu-se por inserir as simulações com os parâmetros de discretização  $\varepsilon_{x\#} = 0.05$  e  $\varepsilon_{x\#} = 0.1$ .

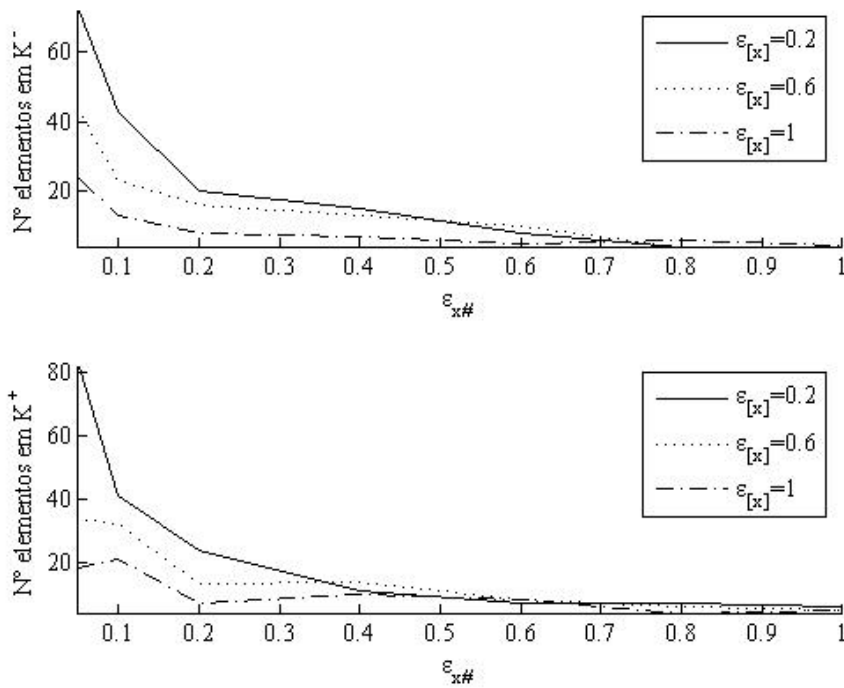


Fig. 4.9: FON - [I]RMOA I:  $\mathbf{K}^-$  e  $\mathbf{K}^+ \times \varepsilon_{x\#}$ . Em ambos os gráficos, o número de pontos nos conjuntos  $\mathbf{K}^-$  e  $\mathbf{K}^+$  diminuem com o número de amostras.

Os gráficos da Fig. 4.10 apresentam que os custos para gerar as amostras e para envelopar  $\mathbf{Y}_d^*$  são relativamente menores a partir de  $\varepsilon_{x\#} = 0.2$ . Conseqüentemente, reduziu-se também a qualidade dos envelopes.

A Fig. 4.11 apresenta duas simulações onde o [I]RMOA I falhou em criar envelopes rigorosos sobre  $\mathbf{Y}_d^*$ . Isso pode ocorrer quando  $\varepsilon_{[x]}$  não é suficientemente menor que  $\varepsilon_{x\#}$  e quando  $w([\mathbf{P}]) = \varepsilon_{[x]}$  (Tab.3.4, linha 5), pois são condições para gerar funções de inclusões estreitas. A discussão sobre este problema encontra-se detalhada no final desta seção.

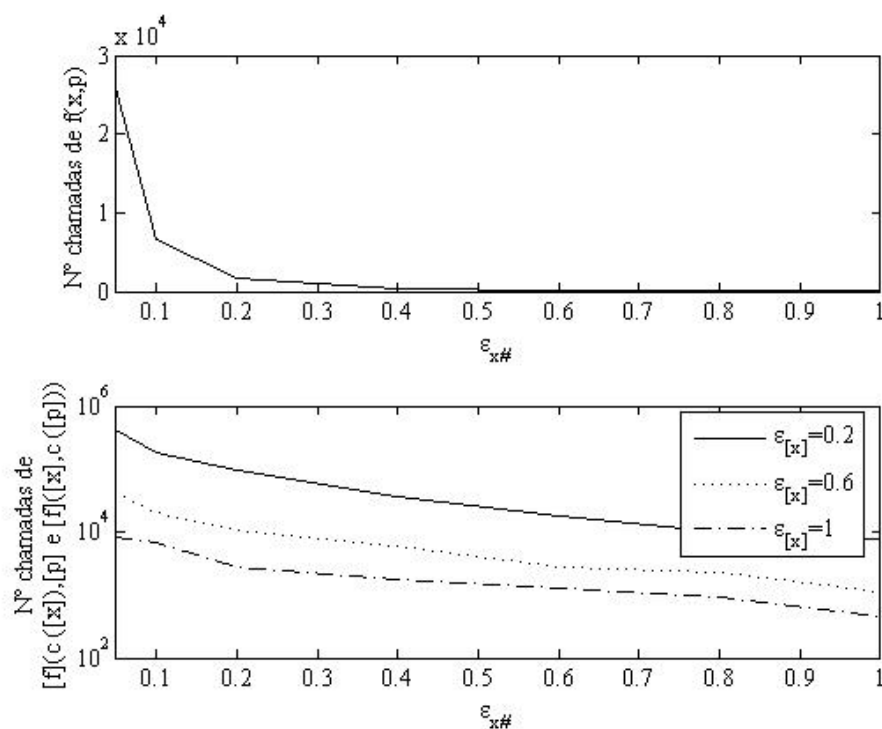
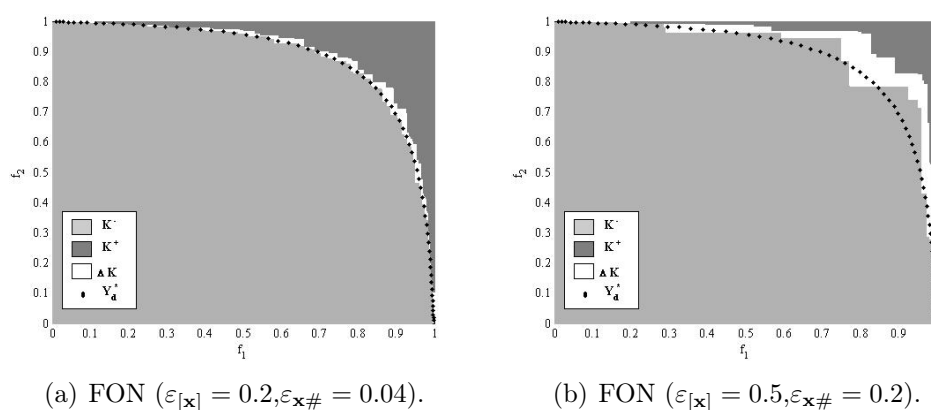


Fig. 4.10: FON - [I]RMOA I: custo computacional. Acima, o esforço na função grade, e a abaixo, o número de chamadas das funções  $f(\mathbf{c}([\mathbf{x}]), [p])$  e  $f([\mathbf{x}], c([p]))$ .



(a) FON ( $\varepsilon_{[\mathbf{x}]} = 0.2, \varepsilon_{\mathbf{x}\#} = 0.04$ ).

(b) FON ( $\varepsilon_{[\mathbf{x}]} = 0.5, \varepsilon_{\mathbf{x}\#} = 0.2$ ).

Fig. 4.11: FON - [I]RMOA I: envelopes sobre  $\mathbf{Y}_d^*$ . A qualidade do envelope sobre  $\mathbf{Y}_d^*$  é melhor para menores valores  $\varepsilon_{[\mathbf{x}]}$  e  $\varepsilon_{\mathbf{x}\#}$ . Em (b), claramente o [I]RMOA I falha, pois em alguns pontos  $\mathbf{Y}^* \subset K^-$ .

[I]RMOA I  $\times$  ZDT1

Para a função ZDT1, foram utilizados  $\varepsilon_{x\#} = [0.02 : 0.01 : 0.1]$  e  $\varepsilon_{[x]} = [0.02 : 0.01 : 0.1]$ . Os resultados encontram-se nas Fig. 4.12, Fig. 4.13 e Fig. 4.14.

Observa-se na Fig. 4.12 que os conjuntos  $\mathbf{K}^-$  e  $\mathbf{K}^+$  contêm relativamente muitos pontos para definição do envelope sobre  $\mathbf{Y}_d^*$ , mesmo para não expressivos valores de  $\varepsilon_{x\#}$ , tais como  $\varepsilon_{x\#} = 0.05$ . Isto ocorre por causa da forma do espaço dos objetivos; ZDT1 tem fronteira robusta convexa e contínua.

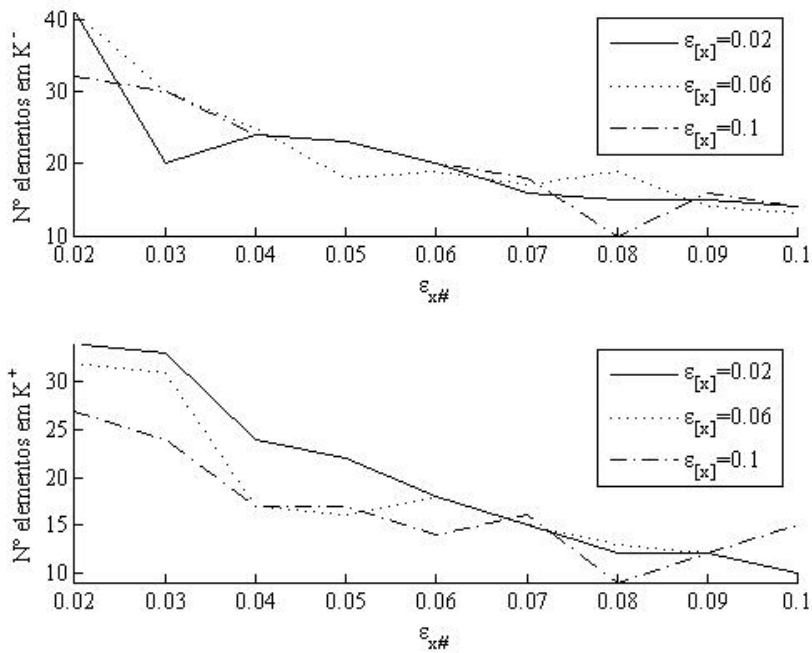


Fig. 4.12: ZDT1 - [I]RMOA I:  $\mathbf{K}^-$  e  $\mathbf{K}^+ \times \varepsilon_{x\#}$ . Em ambos os gráficos, o número de pontos nos conjuntos  $\mathbf{K}^-$  e  $\mathbf{K}^+$  diminuem com o número de amostras com mais regularidade que com as funções SCH2 e FON.

A Fig. 4.13 comprova novamente que o [I]RMOA I exige considerável esforço computacional para classificar os pontos da função grade.

Dois exemplos de envelopes sobre a fronteira robusta estão expostos na Fig. 4.14. Atente para o fato que o [I]RMOA I falha no caso (a), pois parte da fronteira robusta encontra-se sobre a região definida por  $\mathbf{K}^-$ . Isso ocorreu porque a discretização do espaço das incertezas foi insuficiente para resultar funções de inclusão mais estreitas.

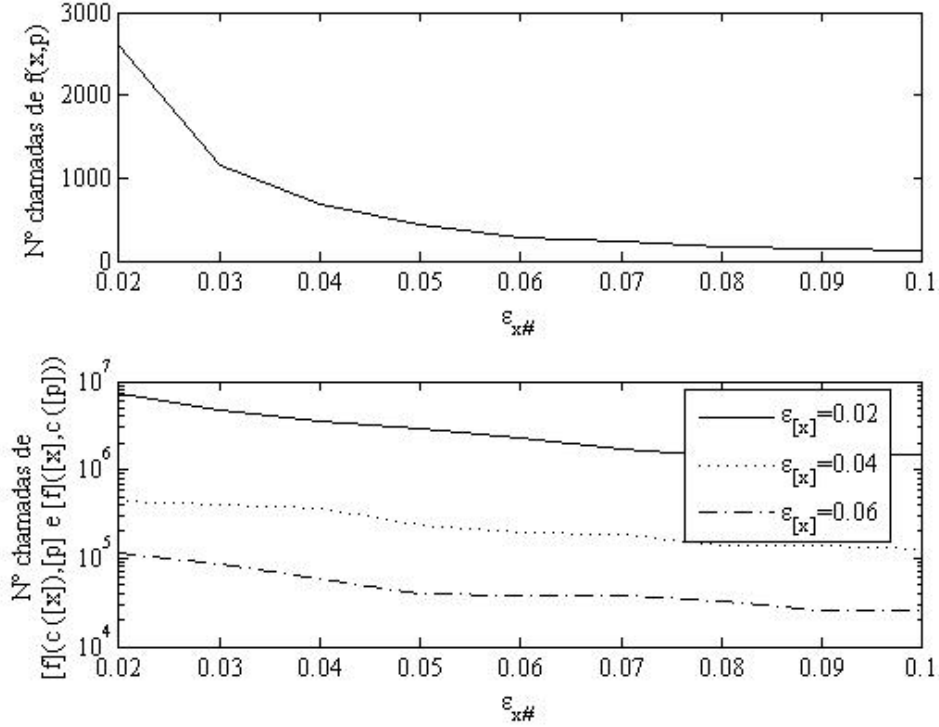


Fig. 4.13: ZDT1 - [I]RMOA I: custo computacional. Acima, o esforço na função grade, e a abaixo, nas funções de inclusão  $[f](\mathbf{c}([\mathbf{x}]), [\mathbf{p}])$  e  $[f](\mathbf{c}([\mathbf{x}], \mathbf{c}([\mathbf{p}])))$ .

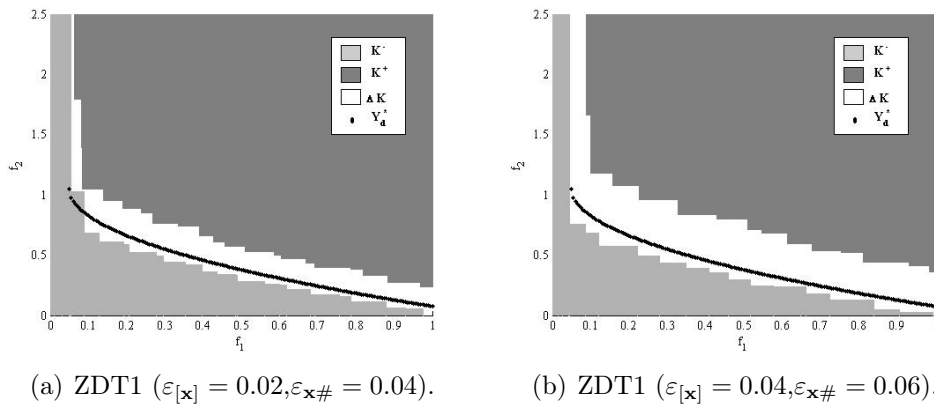


Fig. 4.14: ZDT1 - [I]RMOA I: envelopes sobre  $\mathbf{Y}_d^*$ . Em (a), note que alguns pontos de  $\mathbf{Y}_d^*$  não são cercados pelo envelope  $\Delta K$ . Em (b), o envelope sobre  $\mathbf{Y}_d^*$  é realizado com sucesso.

[I]RMOA I  $\times$  ZDT2

Para a função ZDT2, foram utilizados  $\varepsilon_{x\#} = [0.02 : 0.02 : 0.1]$  e  $\varepsilon_{[x]} = [0.01, 0.02 : 0.02 : 0.1]$ . Os resultados estão apresentados nas figuras Fig. 4.15, Fig. 4.16 e Fig. 4.17.

O número de elementos de  $\mathbf{K}^-$  e  $\mathbf{K}^+$ , veja na Fig. 4.15, seria maior caso não houvesse restrições. Claramente, a separação da fronteira robusta em duas partes pela função de restrição interferiu na definição dos pontos em  $\mathbf{K}^-$  e  $\mathbf{K}^+$  que pertenceriam à região excluída pela restrição.

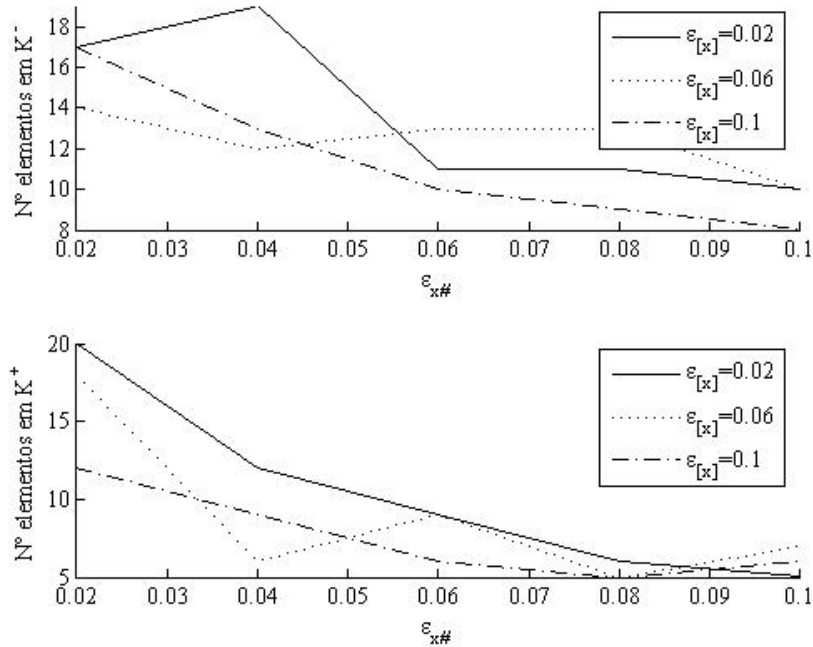


Fig. 4.15: ZDT2:  $\mathbf{K}^-$  e  $\mathbf{K}^+ \times \varepsilon_{x\#}$ . Em ambos gráficos, os trechos onde o número de pontos nos conjuntos  $\mathbf{K}^-$  e  $\mathbf{K}^+$  crescem com o número de amostras de  $\varepsilon_{x\#}$  indicam que a discretização não foi suficientemente adequada.

A Fig. 4.16 apresenta o custo computacional total, ou seja, está contabilizado as consultas às funções de restrição no início do algoritmo.

O tratamento da função de restrição foi realizado anteriormente à discretização, e portanto, nenhum ponto não viável foi gerado no espaço dos objetivos pela função grade. Assim, como esperado, não foi possível visualizar a imagem da região não viável na Fig. 4.17.

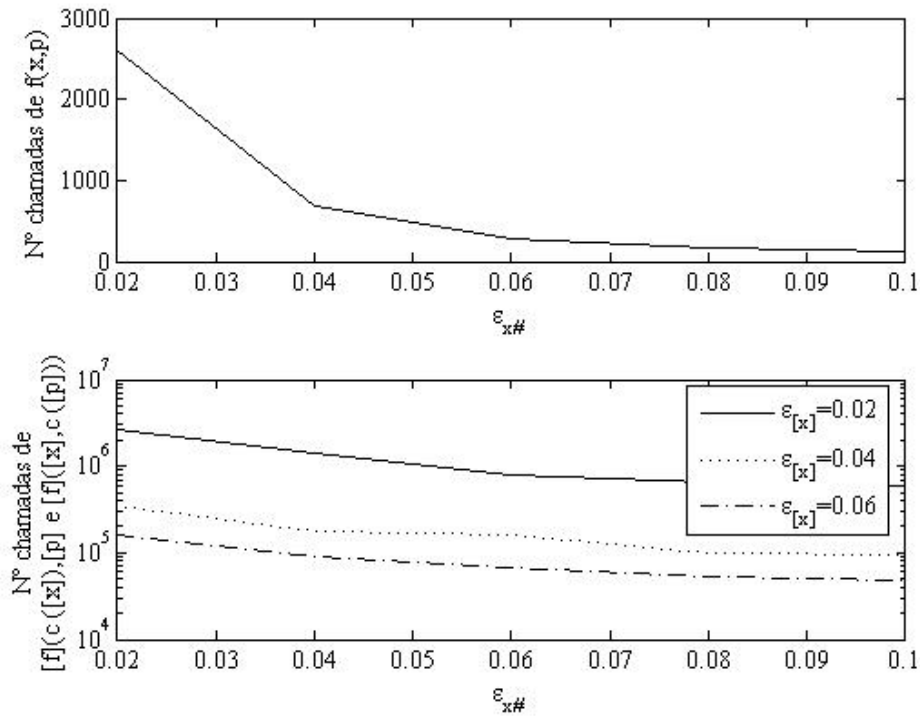
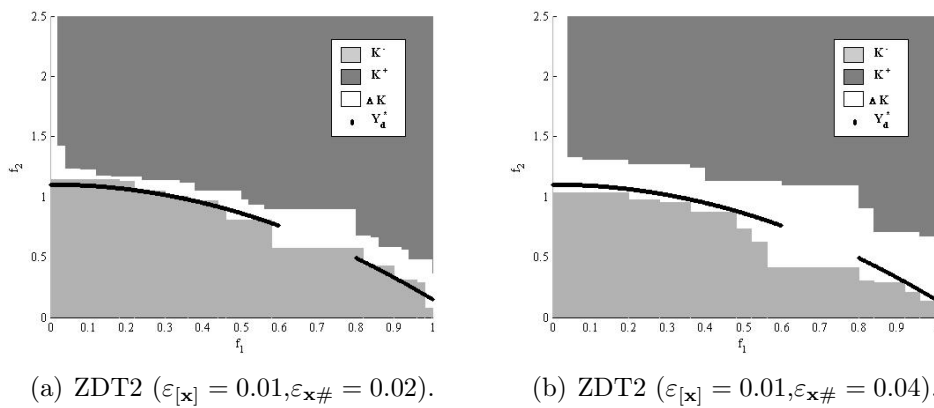


Fig. 4.16: ZDT2: custo computacional. Acima, o esforço da função grade, e a abaixo, o número de chamadas das funções objetivo  $[f](c([x]), [p])$  e  $[f]([x], c([p]))$ .



(a) ZDT2 ( $\epsilon_{[x]} = 0.01, \epsilon_{x\#} = 0.02$ ).

(b) ZDT2 ( $\epsilon_{[x]} = 0.01, \epsilon_{x\#} = 0.04$ ).

Fig. 4.17: ZDT2: envelopes sobre  $Y_d^*$ . Em (a), claramente o [I]RMOA I falha pois, em alguns pontos  $Y^* \subset K^-$ .

[I]RMOA I  $\times$  ZDT3

Para a função ZDT3, foram utilizados  $\varepsilon_{\mathbf{x}\#} = [0.02 : 0.02 : 0.1]$  e  $\varepsilon_{[\mathbf{x}]} = [0.02 : 0.01 : 0.1]$ . Os resultados estão apresentados nas figuras Fig. 4.18, Fig. 4.19 e Fig. 4.20.

Dentre as funções teste, ZDT3 apresentou-se como a mais difícil de envelopar a fronteira robusta. Isso pode ser comprovado pela Fig. 4.18 que mostra o número reduzido de pontos em  $\mathbf{K}^-$  e  $\mathbf{K}^+$ .

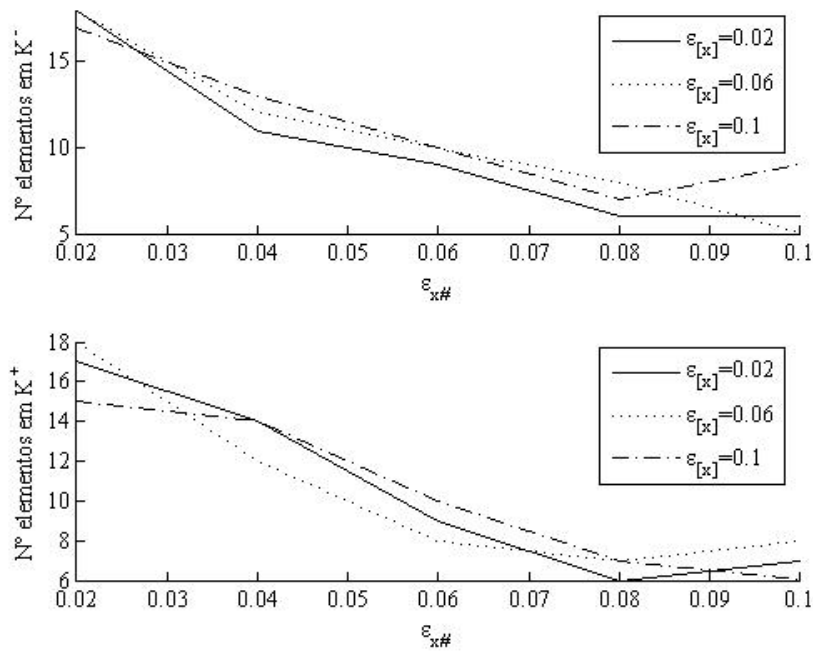


Fig. 4.18: ZDT3:  $\mathbf{K}^-$  e  $\mathbf{K}^+ \times \varepsilon_{\mathbf{x}\#}$ . Em ambos gráficos, o número de pontos nos conjuntos  $\mathbf{K}^-$  e  $\mathbf{K}^+$  diminuem com o número de amostras de  $\varepsilon_{\mathbf{x}\#}$ .

De acordo com a Fig. 4.19, o custo alto para classificar os pontos discretizados também é confirmado para a função ZDT3.

A Fig. 4.20 exhibe dois resultados de envelopes sobre a fronteira robusta. O [I]RMOA I falhou em envelopar  $\mathbf{Y}_d^*$  para os valores de  $\varepsilon_{[\mathbf{x}]}$  e  $\varepsilon_{\mathbf{x}\#}$  testados. Isto é resultado das funções de inclusão  $[\mathbf{f}](\mathbf{c}([\mathbf{x}]), [p])$  e  $[\mathbf{f}](\mathbf{x}, c([p]))$  (Tab.3.4, linhas 3 e 4) não terem sido desenvolvidas suficientemente estreitas. Uma forma de corrigir esse problema é dividir  $[P]$  (Tab.3.4, linhas 5) em subpavimentos  $[p]$  menores ainda.

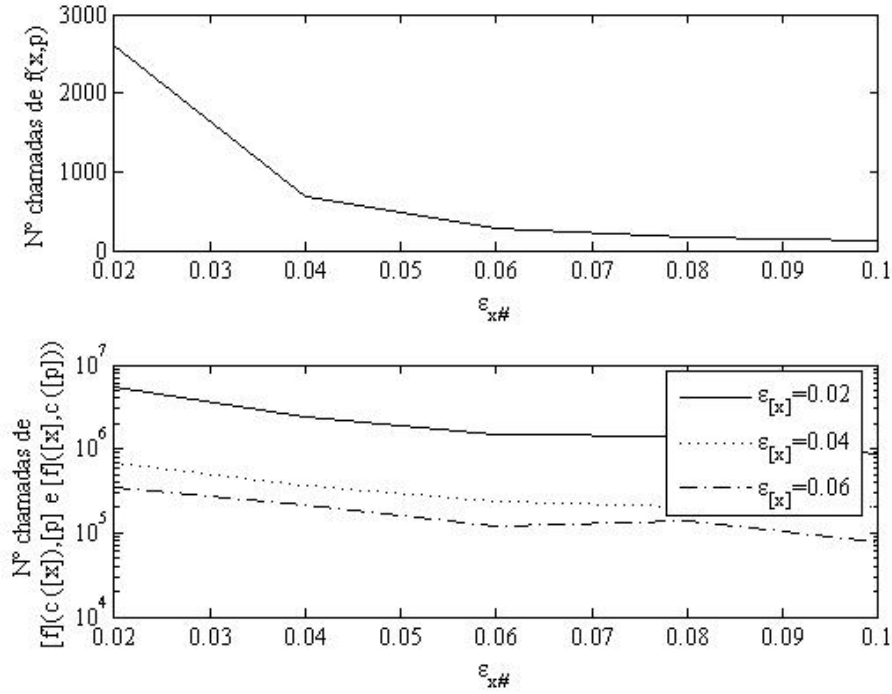


Fig. 4.19: ZDT3: custo computacional. Acima, o esforço da função grade, e a abaixo, o número de chamadas das funções objetivo  $f(c(x), p)$  e  $f(x, c(p))$ .

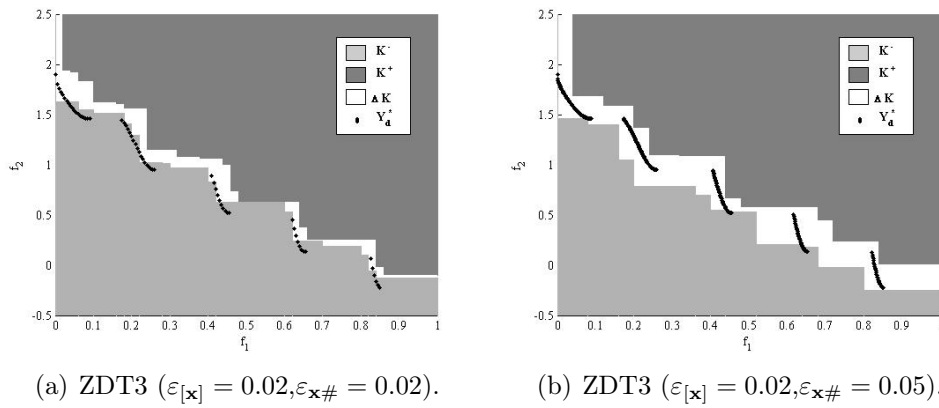


Fig. 4.20: ZDT3: envelopes sobre  $Y_d^*$ . Ambas as tentativas de envelopar  $Y_d^*$  falham, visto que  $Y_d^* \subset K^-$ . O resultado em (b) sugere que os parâmetro  $\epsilon_{[x]}$  deve ser ainda mais preciso para envelopar satisfatoriamente  $Y_d^*$ .

### Considerações sobre o [I]RMOA I

De modo geral, os algoritmos de otimização concentram esforços para encontrar “diretamente” as soluções do problema em questão. Diferentemente, o [I]RMOA I busca por envelopar em  $\Delta\mathbf{K}$  a região no espaço dos objetivos onde a fronteira de Pareto robusta está, por exemplo,  $\Delta\mathbf{K} \subset [\mathbf{Y}]' \setminus (\mathbf{K}^+ \cup \mathbf{K}^-)$  e  $[\mathbf{Y}]'$  como definido em (3.13) e (3.14). A partir dessas informações oferecidas pelo [I]RMOA I, pelo menos dois caminhos podem ser utilizados na seqüência. O primeiro seria computar a região no espaço de busca correspondente à imagem de  $\Delta\mathbf{K}$  empregando-se de algoritmos de projeção de intervalos, tais como o desenvolvido por Dao em [27]. No entanto, para o contexto presente, existem diversas fontes de incerteza que podem prejudicar o resultado final da projeção de  $\Delta\mathbf{K}$ . Por exemplo, geralmente  $[\mathbf{Y}]'$  contém porções extras de espaço (veja Fig. 3.1) e essas regiões também teriam sua projeção realizada. Neste caso, o resultado da projeção deveria ainda ser submetido à intersecção com o espaço de busca inicial, para eliminar os espaços extras. O segundo caminho seria considerar o [I]RMOA I como um método de validação. Em outras palavras, ele seria utilizado para auxiliar o projetista/decisor a examinar outros algoritmos robustos. Veja como. A precisão continuaria sendo estabelecida pelo parâmetro  $\varepsilon_{\mathbf{x}\#}$  e a fronteira robusta do método em teste  $\mathbf{Y}_s^*$  deveria obedecer à seguinte inclusão  $\mathbf{Y}_s^* \subset [\mathbf{Y}]'/(\mathbf{K}^- \cup \mathbf{K}^+)$ .

O [I]RMOA I é caro computacionalmente. O processo interno que mais depende esforço é o FPS (Tab.3.3) e CSC (Tab.3.4). Esse processo se baseia na bissecção do espaço de busca e de incertezas para definir se um ponto pertence a  $\mathbf{K}^-$  ou  $\mathbf{K}^+$ . Logo, a dimensionalidade desses espaços pode inviabilizar a aplicação do método. A qualidade dessa busca, e conseqüentemente o custo, dependem dos parâmetros  $\varepsilon_{[\mathbf{x}]}$ ,  $\varepsilon_{\mathbf{x}\#}$ , da discretização do parâmetro de incerteza  $[\mathbf{P}]$  e da qualidade das funções de inclusão. O parâmetro  $\varepsilon_{[\mathbf{x}]}$  deve ser suficientemente menor que  $\varepsilon_{\mathbf{x}\#}$ , pelo menos  $\varepsilon_{[\mathbf{x}]} < \varepsilon_{\mathbf{x}\#}/2$ , para garantir que a precisão de busca seja melhor que a precisão da discretização. Além disso, quanto mais subpavimentos de  $[\mathbf{P}]$ , mais estreita e mais precisa será a função de inclusão sobre os pontos solução (veja Fig. 3.4). Em outras palavras, subpavimentos  $[\mathbf{p}]$  maiores que o adequado podem conduzir a erros de precisão e conseqüentemente a erros na computação de pertinência das soluções candidatas às regiões  $\mathbf{K}^-$  e  $\mathbf{K}^+$ .

Portanto, quando utilizou-se  $n_{sp}$  adequado para cada teste, fronteira robusta discreta  $\mathbf{Y}_s^*$  foi envelopada pelo [I]RMOA I. Considerando o quadro acima e as funções teste definidas anteriormente, o [I]RMOA I foi declarado aprovado.

## 4.3.2 [I]RMOA II

O método [I]RMOA II não precisa discretizar o espaço de busca previamente como ocorre no [I]RMOA I. A discretização é realizada durante o processo de otimização, utilizando-se de um parâmetro de precisão para o espaço de busca  $\varepsilon_{[x]}$  e um parâmetro de precisão para o espaço de incertezas  $\varepsilon_{[p]}$ . Nos testes realizados,  $\varepsilon_{[p]}$  foi mantido constante, e portanto, o comportamento do [I]RMOA II foi examinado variando-se  $\varepsilon_{[x]}$ . Para cada função teste três tipos de resultados estão apresentados: a) o número de soluções em relação à  $\varepsilon_{[x]}$  juntamente com o esforço computacional medido em número de cálculo de funções objetivo; b) a comparação do conjunto solução  $\mathbf{X}_s^*$  com o conjunto solução discreto  $\mathbf{X}_d^*$ ; e c) a comparação da imagem do conjunto solução  $\mathbf{Y}_s^*$  em relação a fronteira computada  $\mathbf{Y}_d^*$ .

O valor dos parâmetros  $\varepsilon_{[x]}$  e  $\varepsilon_{[p]}$  empregados durante os testes desta subseção foram listados.

Tab. 4.2: [I]RMOA II - Parâmetros de testes.

Fonction	$\varepsilon_{[x]}$	$\varepsilon_{[p]}$
SCH2	[0.1 : 0.1 : 1]	0.2
FON	[0.1 : 0.1 : 1]	0.025
ZDT1	[0.01 : 0.01 : 0.1]	0.02
ZDT2	[0.01 : 0.01 : 0.1]	0.02
ZDT3	[0.01 : 0.01 : 0.1]	0.02

[I]RMOA II  $\times$  SCH2

Para a função SCH2, o parâmetro de precisão examinado foi  $\varepsilon_{[x]} = [0.1 : 0.1 : 1]$ . Os resultados estão apresentados na Fig. 4.21, Fig. 4.22 e Fig. 4.23.

O número de soluções encontradas pelo [I]RMOA II e o esforço computacional estão mostrados na Fig. 4.21. Observe que, de forma geral, o número de soluções decresce com a diminuição da precisão. No entanto, em alguns trechos o número de soluções permanece inalterado. Nesses trechos, como por exemplo entre  $\varepsilon_{[x]} = 0.5$  e  $\varepsilon_{[x]} = 0.9$ , esse comportamento é justificado pois os parâmetros de precisão  $\varepsilon_{[x]}$  são equivalentes em termos de precisão.

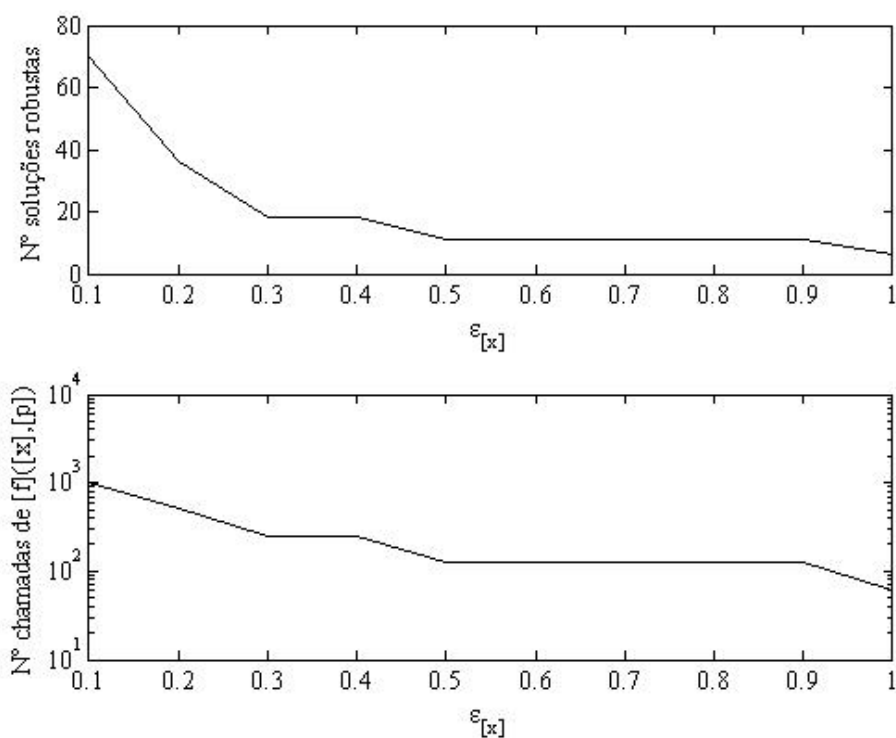


Fig. 4.21: SCH2 - [I]RMOA II: quantitativo de soluções robustas e de esforço computacional segundo o parâmetro de precisão  $\varepsilon_{[x]}$ .

A precisão do método e a uniformidade dos conjuntos solução podem ser examinadas na Fig. 4.22. Nota-se que não é necessário utilizar a métrica TE, pois os conjuntos solução estão em concordância com a proposta de precisão definida pelo parâmetro  $\varepsilon_{[x]}$ . A distribuição do conjunto solução é claramente homogênea.

A Fig. 4.23 compara  $\mathbf{Y}_d^*$  com  $\mathbf{Y}_s^*$ . Observa-se que o resultado cobre regularmente a fronteira robusta. No entanto, o espaçamento entre as imagens

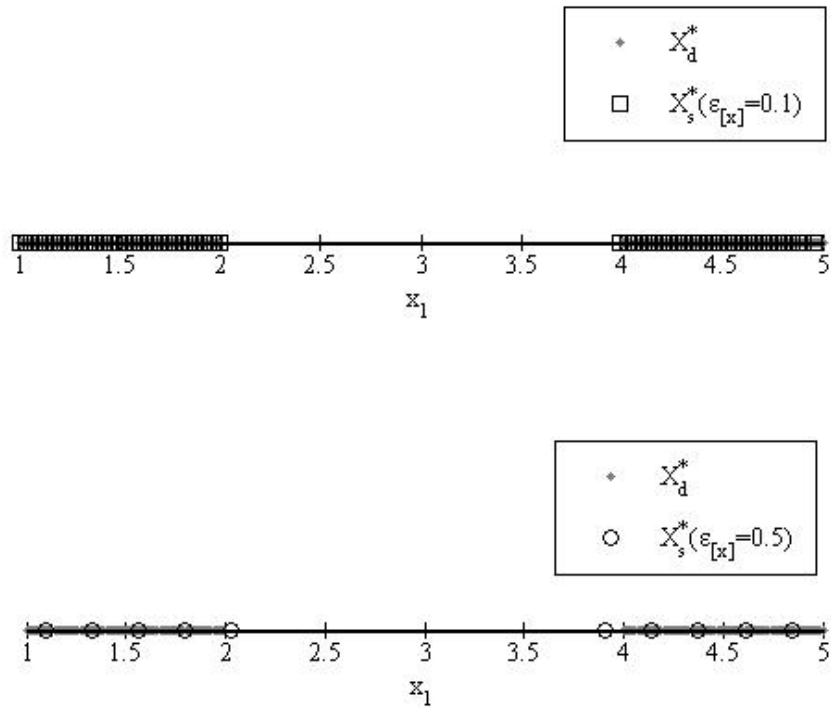
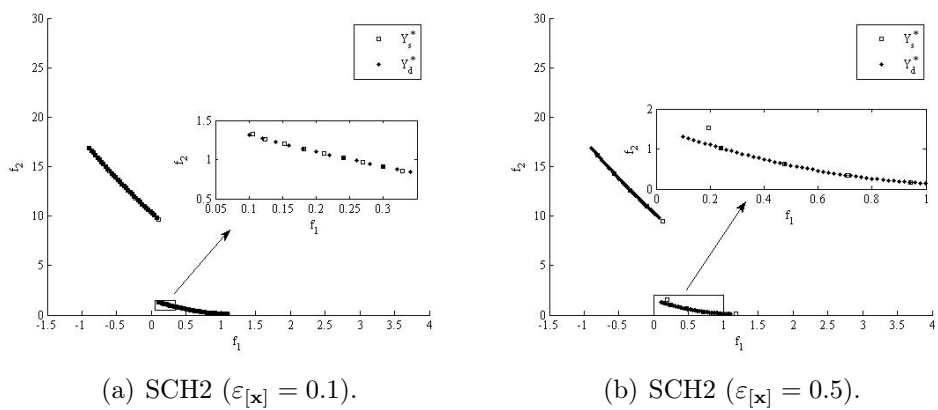


Fig. 4.22: SCH2 - [I]RMOA II: paralelo entre  $\mathbf{X}_d^*$  e  $\mathbf{X}_s^*$ .

das soluções encontradas  $\mathbf{Y}_s^*$  é devido à precisão utilizada em cada simulação.



(a) SCH2 ( $\varepsilon_{[x]} = 0.1$ ).

(b) SCH2 ( $\varepsilon_{[x]} = 0.5$ ).

Fig. 4.23: SCH2 - [I]RMOA II: paralelo entre  $\mathbf{Y}_d^*$  e  $\mathbf{Y}_s^*$ .

[I]RMOA II  $\times$  FON

Para a função FON, o parâmetro de precisão examinado foi  $\varepsilon_{[x]} = [0.1 : 0.1 : 1]$ . Os resultados estão apresentados nas figuras Fig. 4.24, Fig. 4.25 e Fig. 4.26.

O processo de busca do [I]RMOA II mostrou-se independente da não convexidade da função FON. Na verdade, esse processo de busca é somente vinculado aos parâmetros de precisão  $\varepsilon_{[x]}$  e  $\varepsilon_{[p]}$ . O número de soluções robustas e o esforço computacional estão mostrados na Fig. 4.24.

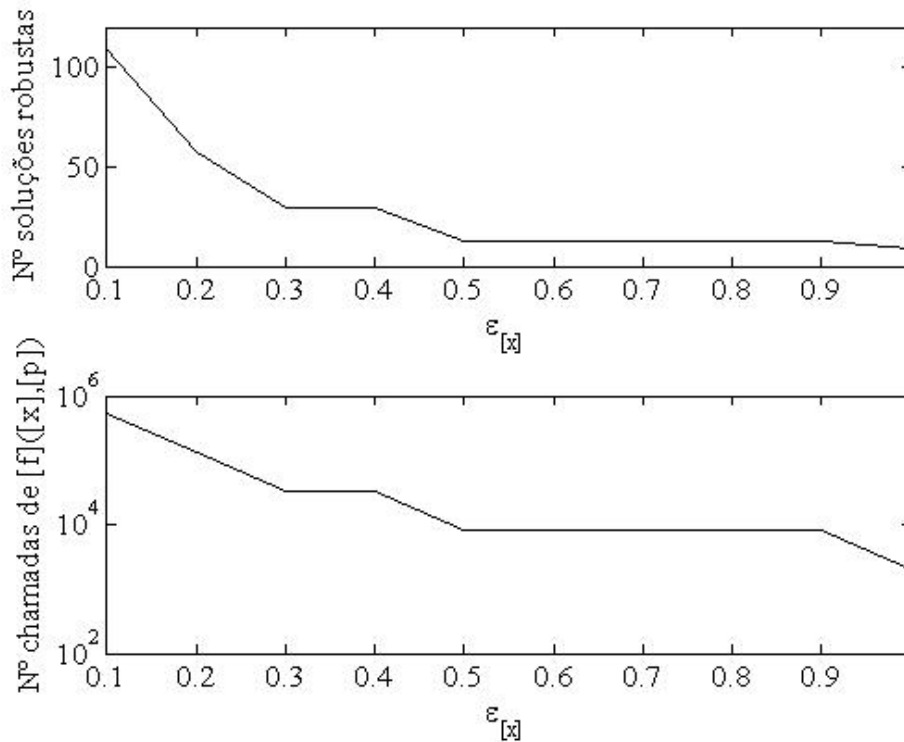


Fig. 4.24: FON - [I]RMOA II: quantitativo de soluções robustas e de esforço computacional segundo o parâmetro de precisão  $\varepsilon_{[x]}$ .

A Fig. 4.25 compara os conjuntos solução do [I]RMOA II para duas instâncias de  $\varepsilon_{[x]}$ . Visualmente não está claro se a precisão em cada caso foi atendida. Por isso, aplicou-se a métrica TE, segundo a precisão proposta. O resultado foi de 100% de conformidade dos conjuntos  $\mathbf{X}_s^*$  em relação a  $\mathbf{X}_d^*$ .

A imagem de dois conjuntos solução está mostrada na Fig. 4.26. Note que o espaçamento não uniforme no espaço de busca foi substituído por um espaçamento homogêneo no espaço dos objetivos. Esse resultado é devido a ação da técnica de nicho NB[I] no espaço dos objetivos.

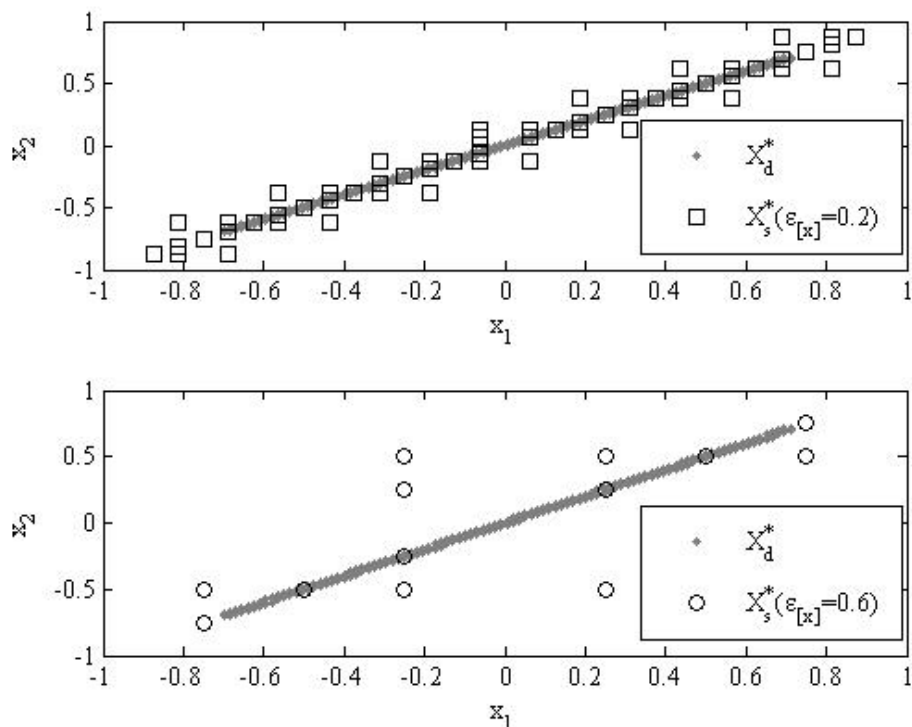


Fig. 4.25: FON - [I]RMOA II: paralelo entre  $\mathbf{X}_d^*$  e  $\mathbf{X}_s^*$ . Percebe-se a aproximação ao conjunto solução discreto  $\mathbf{X}_d^*$  a medida que diminui-se  $\varepsilon_{[x]}$ .

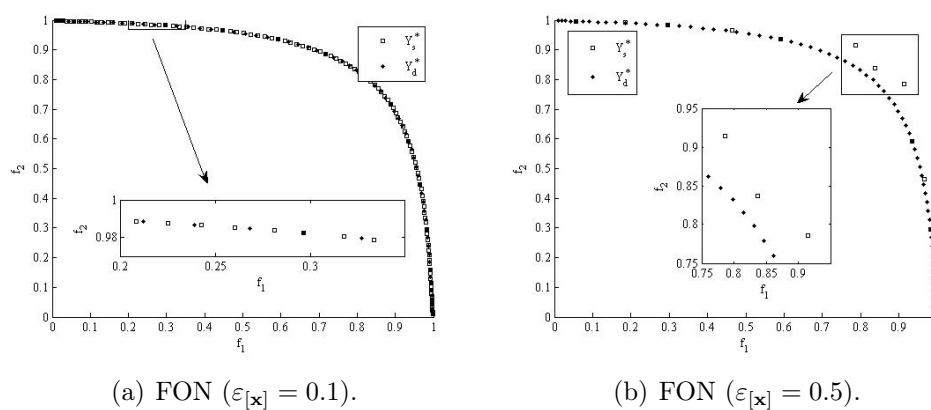


Fig. 4.26: FON - [I]RMOA II: paralelo entre  $\mathbf{Y}_d^*$  e  $\mathbf{Y}_s^*$ . Os gráficos contêm resultados em conformidade com a precisão proposta. Entretanto, (a) apresenta um resultado mais interessante:  $\mathbf{Y}_s^*$  no mesmo alinhamento de  $\mathbf{Y}_d^*$ .

[I]RMOA II  $\times$  ZDT1

Para a função ZDT1, o parâmetro de precisão utilizado foi  $\varepsilon_{[x]} = [0.01 : 0.01 : 0.1]$ . O custo computacional e o número de elementos de  $\mathbf{X}_s^*$  estão apresentados na Fig. 4.27.

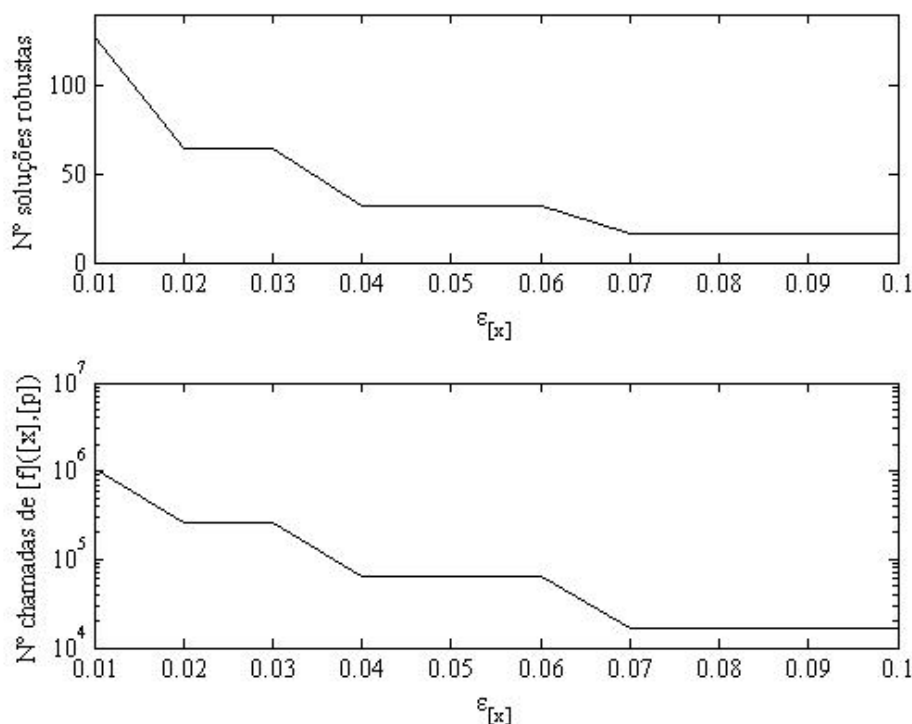


Fig. 4.27: ZDT1 - [I]RMOA II: quantitativo de soluções robustas e de esforço computacional segundo o parâmetro de precisão  $\varepsilon_{[x]}$ .

A Fig. 4.28 confronta  $\mathbf{X}_d^*$  com vários  $\mathbf{X}_s^*$ . O exame visual da figura é suficiente para avaliar a precisão e uniformidade de espaçamento dos conjuntos solução. O número de soluções robustas para  $\varepsilon_{[x]} = 0.05$  e  $\varepsilon_{[x]} = 0.1$  é contável diretamente da figura; para  $\varepsilon_{[x]} = 0.01$  e demais valores de precisão, veja a Fig. 4.27.

A Fig. 4.29 mostra a fronteira gerada pelo [I]RMOA II em duas simulações. Ambas as fronteiras são homogêneas em espaçamento e respeitam o valor de  $\varepsilon_{[x]}$  estipulado como precisão do método.

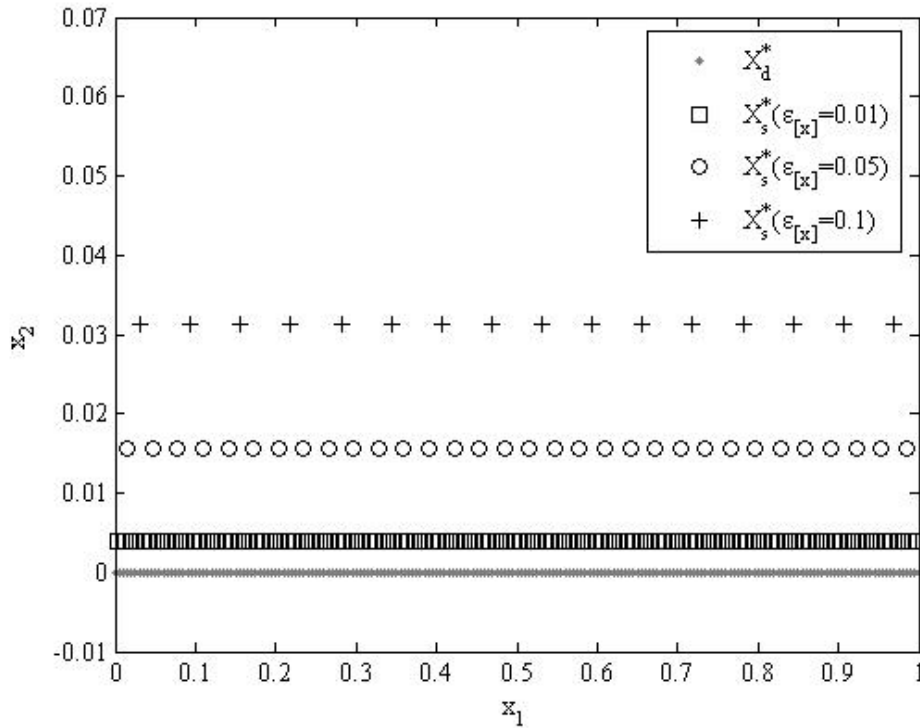
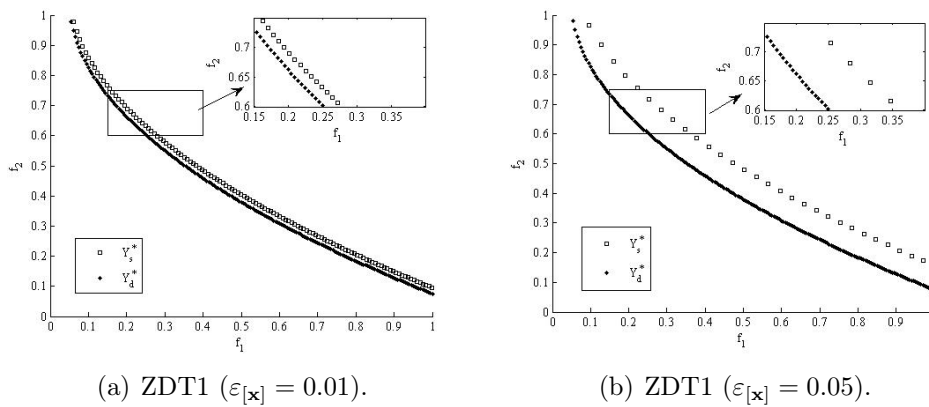


Fig. 4.28: ZDT1 - [I]RMOA II: paralelo entre  $\mathbf{X}_d^*$  e  $\mathbf{X}_s^*$ . Observa-se que todos os conjuntos solução estão em conformidade como o valor de precisão proposto.



(a) ZDT1 ( $\varepsilon_{[x]} = 0.01$ ).

(b) ZDT1 ( $\varepsilon_{[x]} = 0.05$ ).

Fig. 4.29: ZDT1 - [I]RMOA II: paralelo entre  $\mathbf{Y}_d^*$  e  $\mathbf{Y}_s^*$ . Nota-se que a convergência para fronteira  $\mathbf{Y}_d^*$  depende nitidamente do parâmetro de precisão.

[I]RMOA II  $\times$  ZDT2

Para a função ZDT2, o parâmetro de precisão analisado foi  $\varepsilon_{[x]} = [0.01 : 0.01 : 0.1]$ . Considerando-se inicialmente as questões de custo e de número de soluções gerados, comparou-se os resultados para função ZDT1 (Fig. 4.27) com os resultados para ZDT2 (Fig. 4.30). Em valores numéricos, as duas simulações produziram resultados similares confirmando que a característica de convexidade não interfere nos processos de busca do [I]RMOA II. A semelhança entre os produtos das simulações seria ainda maior caso a função ZDT2 fosse irrestrita.

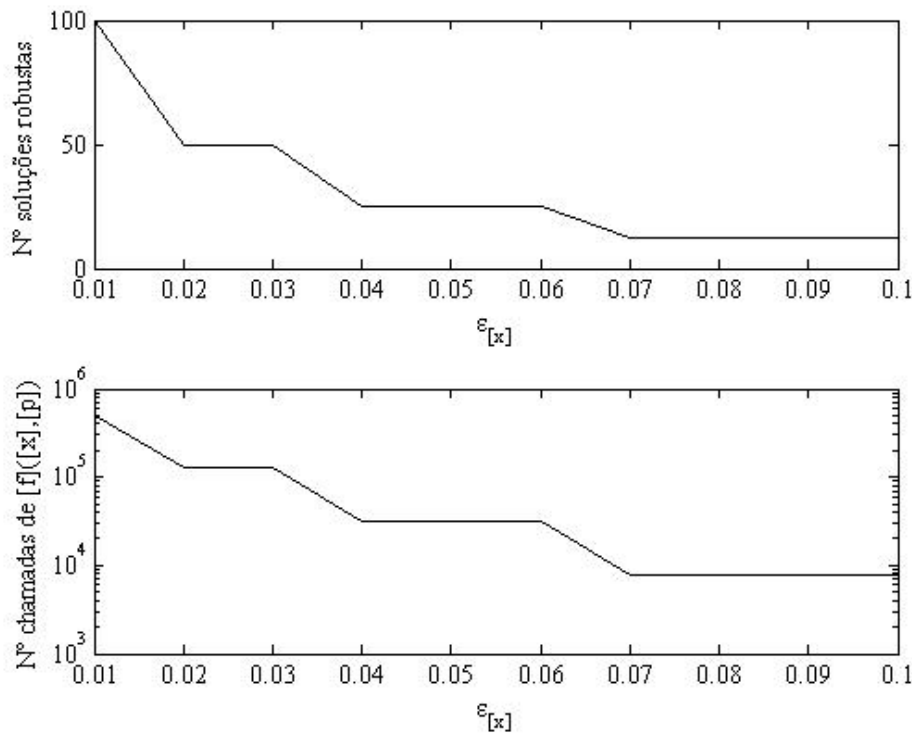


Fig. 4.30: ZDT2 - [I]RMOA II: quantitativo de soluções robustas e de esforço computacional segundo o parâmetro de precisão  $\varepsilon_{[x]}$ .

A Fig. 4.31 mostra os produtos do [I]RMOA II no espaço de busca. Atenta-se para o fato de que  $\varepsilon_{[x]}$  interfere no mapeamento de pontos próximos à função de restrição. Veja na Seção 3.5.1 como as funções de restrição são tratadas para mais detalhes.

As imagens de dois conjuntos solução estão plotadas na Fig. 4.32. Novamente, nem a não convexidade da função e nem a restrição são empecilhos para o [I]RMOA II encontrar as soluções robustas.

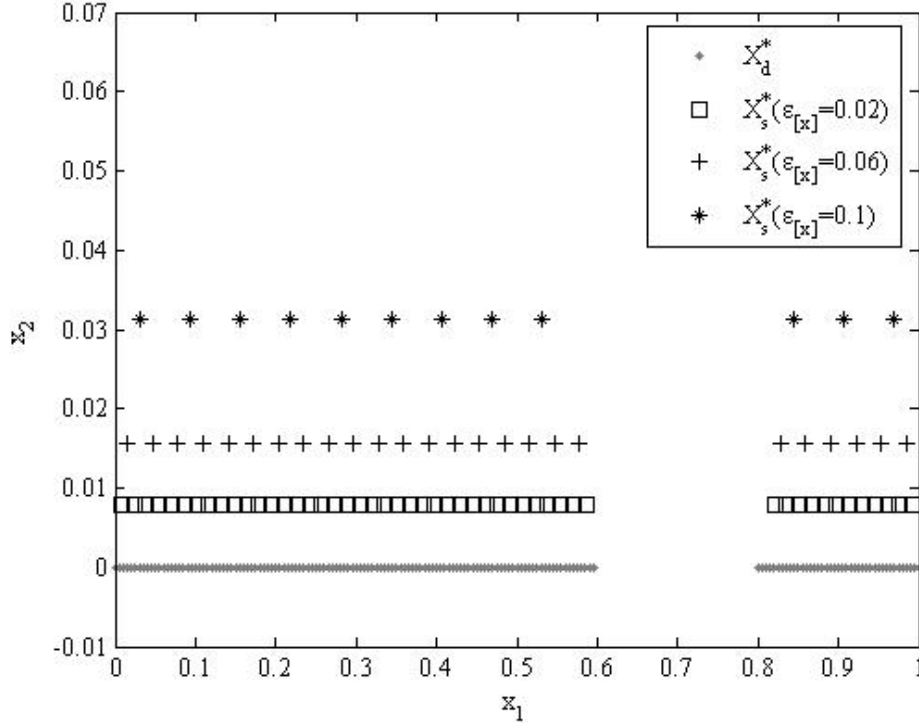


Fig. 4.31: ZDT2 - [I]RMOA II: paralelo entre  $\mathbf{X}_d^*$  e  $\mathbf{X}_s^*$ . Percebe-se que nenhum ponto está presente na região não viável, pois somente são gerados pontos que respeitaram a função de inclusão de restrição.

### [I]RMOA II $\times$ ZDT3

Para a função ZDT3, o parâmetro de precisão examinado foi  $\varepsilon_{[x]} = [0.01 : 0.01 : 0.1]$ . Os resultados encontram-se nas Fig. 4.30, Fig. 4.31 e Fig. 4.35.

O custo computacional é similar às demais funções da família ZDT. No entanto, a descontinuidade da fronteira robusta diminuiu a região efetiva da fronteira robusta. Logo, muito menos soluções foram encontradas para mesmo valor de precisão.

A uniformidade também pode ser verificada na função ZDT3, veja na Fig. 4.35, mesmo com menos elementos em cada região ótima.

As imagens de dois conjuntos robustos estão plotadas Fig. 4.35. As comparações entre os conjuntos  $\mathbf{Y}_d^*$  e  $\mathbf{Y}_s^*$  confirmam que o [I]RMOA II garante encontrar a fronteira, segundo a precisão indicada.

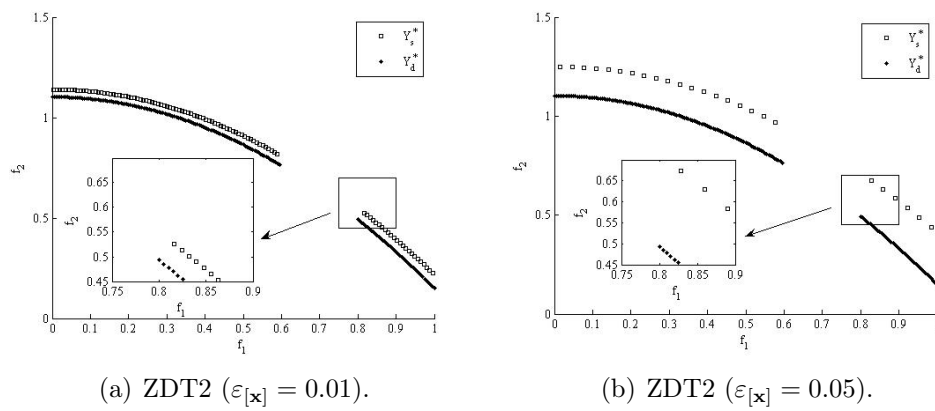


Fig. 4.32: ZDT2 - [I]RMOA II: paralelo entre  $\mathbf{Y}_d^*$  e  $\mathbf{Y}_s^*$ . A uniformidade de espaçamento entre as soluções e a precisão não foram prejudicados pela presença de restrições.

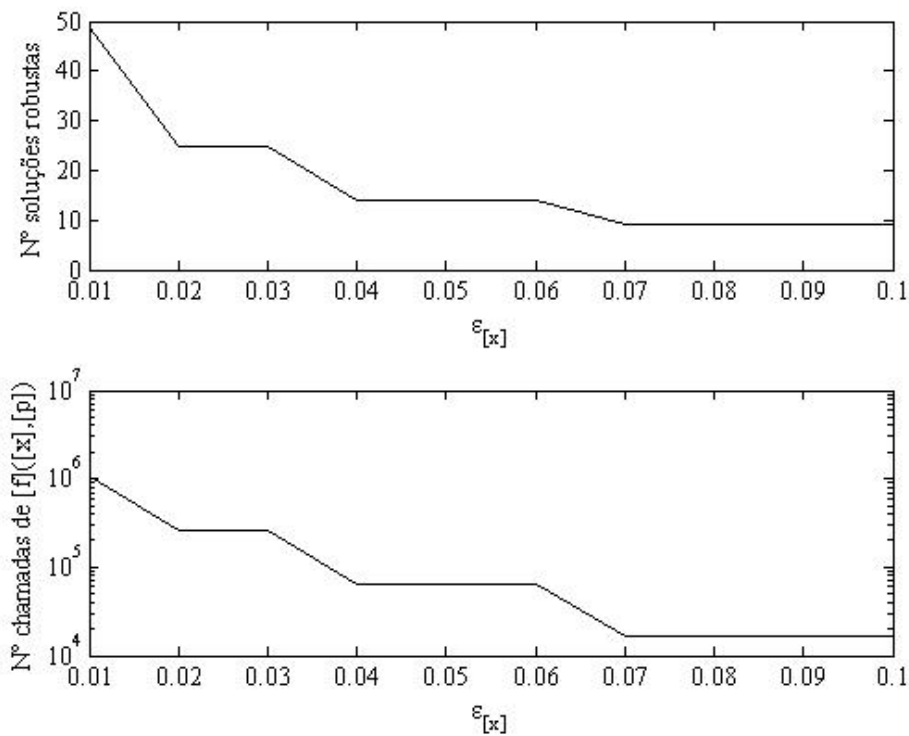


Fig. 4.33: ZDT3 - [I]RMOA II: quantitativo de soluções robustas e de esforço computacional segundo o parâmetro de precisão  $\varepsilon_{[x]}$ .

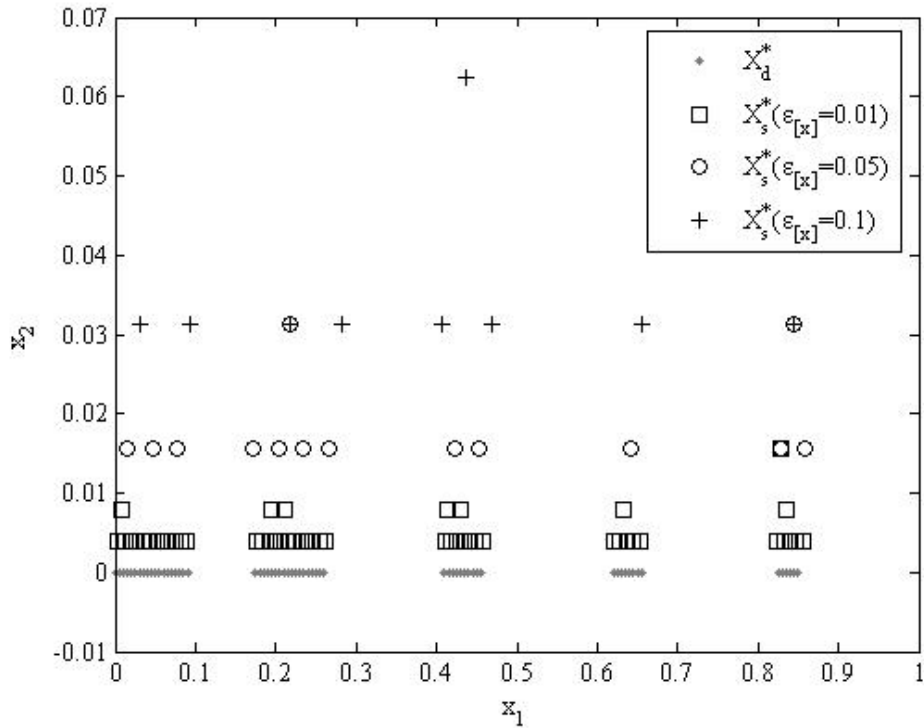
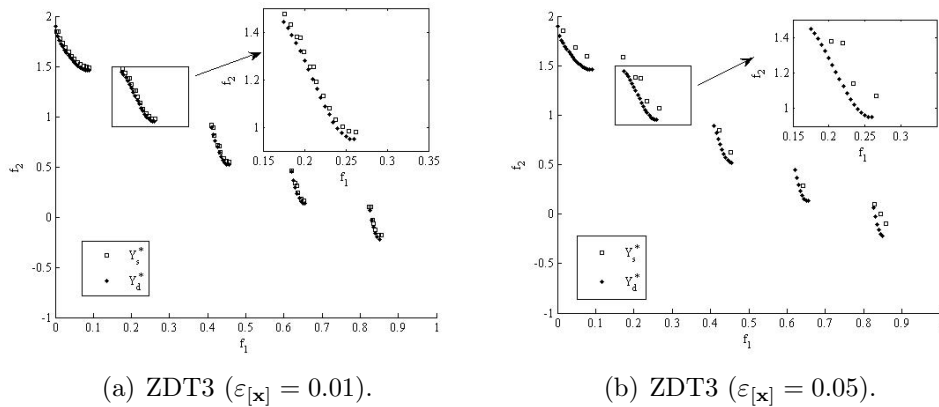


Fig. 4.34: ZDT3 - [I]RMOA II: paralelo entre  $\mathbf{X}_d^*$  e  $\mathbf{X}_s^*$ .



(a) ZDT3 ( $\varepsilon_{[x]} = 0.01$ ).

(b) ZDT3 ( $\varepsilon_{[x]} = 0.05$ ).

Fig. 4.35: ZDT3 - [I]RMOA II: paralelo entre  $\mathbf{Y}_d^*$  e  $\mathbf{Y}_s^*$ .

### Considerações sobre o [I]RMOA II

O número de soluções robustas e o custo para encontrá-las dependem inversamente do valor dos parâmetros de precisão  $\varepsilon_{[x]}$  e  $\varepsilon_{[p]}$ .

De modo geral, os métodos intervalares buscam resolver problemas a que são propostos eliminando regiões onde a solução seguramente não está. As regiões restantes são normalmente armazenadas em listas de caixas intervalares, cuja quantidade de elementos é inversamente proporcional ao valor do parâmetro de precisão do método. O [I]RMOA II segue essa regra geral dos métodos intervalares, não descartando porções do espaço onde possa existir solução robusta, ou seja, o [I]RMOA II emprega estratégia pessimista (veja Fig.2.8). Por isso ele é considerado um método garantido. No entanto, existe um custo para armazenar e classificar as regiões não excluídas. A cada iteração essas regiões não excluídas são bisseccionadas, e portanto o trabalho para prosseguir nos processos de otimização é aumentado. O problema se agrava quando o espaço de busca e de incertezas é multi-dimensional. Se a memória de máquina disponível for considerada suficiente e tempo computacional for um fator menos importante, o [I]RMOA II converge globalmente para todas as soluções.

Uniformidade. As figuras dessa seção destinadas à visualização das fronteiras robustas ou aquelas que apresentam os conjuntos de minimizadores exibem resultados claramente uniformes seja no espaço de busca seja no espaço dos objetivos. Isso devido a três razões simples: a) a bissecção é realizada por corte no ponto médio do primeiro intervalo com maior largura da caixa, resultando sempre em duas caixas simétricas; b) a cada iteração todas as caixas são bisseccionadas, resultando em um conjunto de caixas de mesma forma e volume; e c) o procedimento de não exclusão de espaços com solução, pois isso resulta na não existência de lacunas na fronteira robusta. Em conjunto, esses fatores resultam na convergência para fronteira robusta de forma gradual, uniforme e garantida. No entanto, a uniformidade pode ser corrompida nos problemas restritos, visto que o espaço de busca viável é formado, normalmente, por caixas de forma e volume não uniformes que passaram pelos testes das funções de restrição.

### 4.3.3 [I]RMOEA

Os algoritmos evolucionários e outras técnicas estocásticas de busca produzem resultados diferentes a cada nova execução, pois seus processos internos são guiados por operadores probabilísticos. Geralmente, são utilizados vários testes para estudar o comportamento desses métodos frente à mudança de seus operadores evolucionários. Especificamente, o [I]RMOEA foi estudado fixando-se alguns operadores e examinando-se os outros que mais influenciaram no processo de busca.

A avaliação do [I]RMOEA foi realizada fixando-se a probabilidade dos parâmetros genéticos  $p_c = 1$  e  $p_m = 0.001$  e variando-se os parâmetros  $n_{pop}$  e

$n_{gen}$  para cada caso de teste. Cada indivíduo foi codificado com  $n_{bit} = 10$  bits e para técnica de nicho empregou-se  $r_{bis}^{max} = 8$  rodadas de bissecções. Para análise do [I]RMOEA, cinco tipos de resultados estão apresentados para cada função teste: a) a convergência do algoritmo medida em número de fronteiras em relação à  $n_{gen}$ , para diversos valores de  $n_{pop}$ ; b) a métrica TE em relação à  $n_{pop}$ , para diversos valores de  $n_{gen}$ ; c) a métrica MB[I] em relação à  $n_{pop}$ , para diversos valores de  $n_{gen}$ ; d) a imagem do conjunto solução  $\mathbf{Y}_s^*$  em relação a fronteira computada  $\mathbf{Y}_d^*$ ; e) o esforço computacional por geração, medido em número de cálculo de funções objetivo, em relação à  $n_{pop}$ . Os itens (a-d) foram apresentados separadamente por função teste e, o item (e) está apresentado no gráfico da Fig.4.36.

Considerando o número de cálculos de funções objetivo, o esforço computacional (Fig. 4.36) no [I]RMOEA possui custo fixo com número de cálculos de função proporcional ao tamanho da população, sendo a constante de proporcionalidade o número de subpavimentos originados da discretização do espaço das incertezas (Tab.3.8).

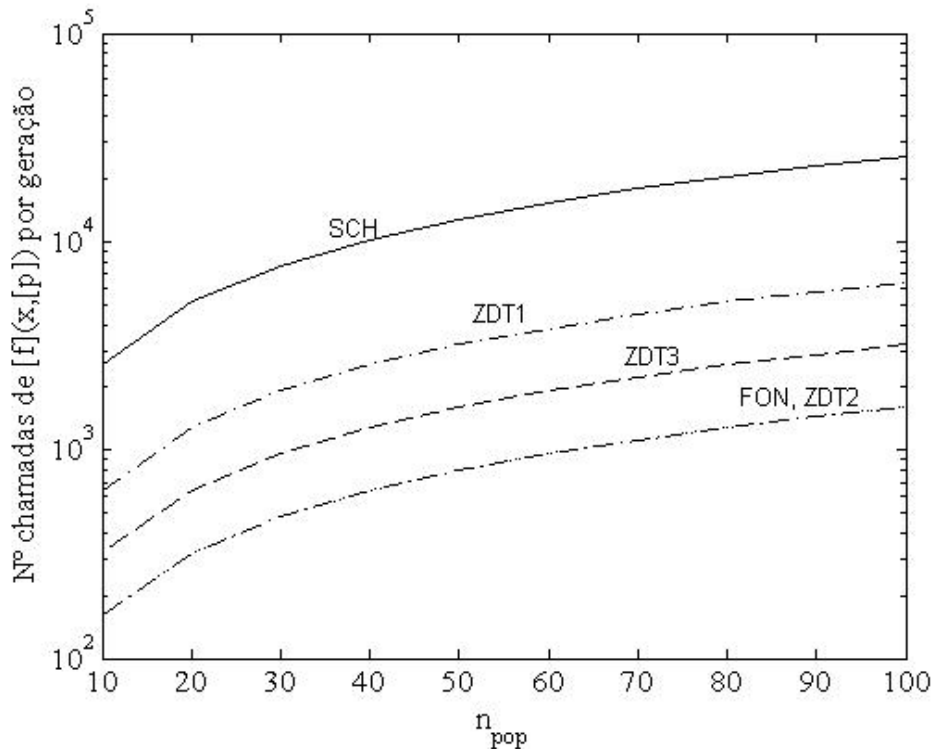


Fig. 4.36: [I]RMOEA: Custo computacional por geração. Observa-se que não foi considerado o custo da função de restrição para a função ZDT2.

[I]RMOEA  $\times$  SCH2

Para a função SCH2, utilizou-se  $n_{pop} = [10 : 10 : 100]$  e  $n_{gen} = [10 : 10 : 50]$ . A Fig. 4.37 apresenta a convergência do [I]RMOEA ao longo das gerações. Por se tratar de uma função com uma única variável e de simples convergência, o número de fronteiras diminuiu rapidamente já nas primeiras gerações, mesmo para as populações com maior número de indivíduos. Se fosse desejado, a convergência poderia ser desacelerada aumentando-se, por exemplo, a probabilidade de mutação  $p_m$ .

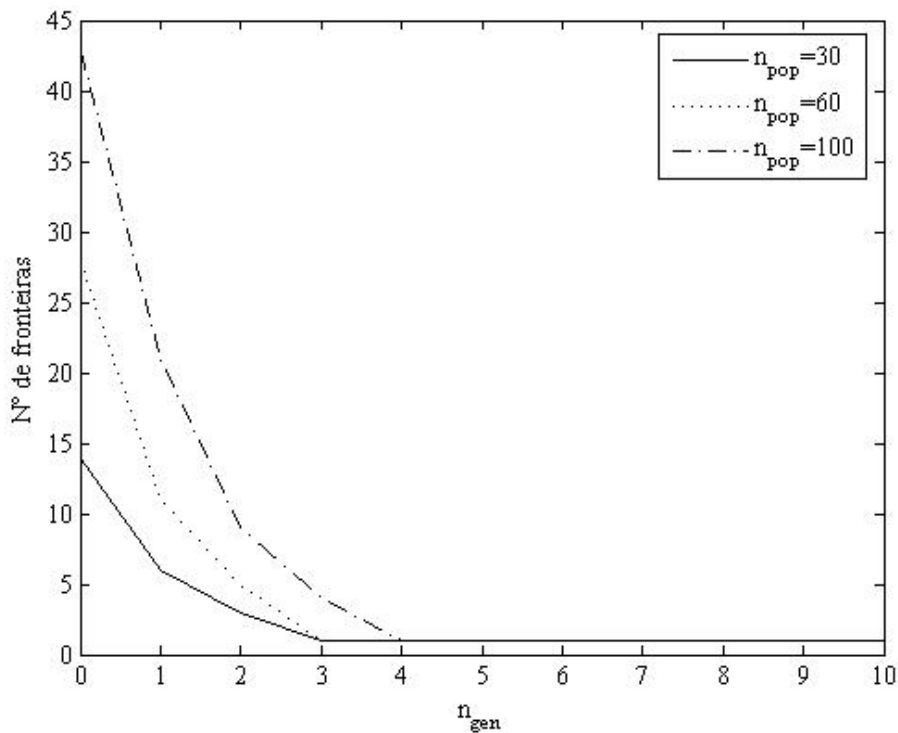


Fig. 4.37: SCH - [I]RMOEA: Convergência das fronteiras. O número de fronteiras decresce indicando que o algoritmo converge para sua primeira fronteira.

A Fig. 4.38 apresenta a eficiência do [I]RMOEA em encontrar o conjunto robusto, medida em termos de TE. A métrica TE, utilizada com precisão  $\varepsilon_{TE} = 0.1$ , acusou 0% de erro, ou seja, todas as soluções foram classificadas como pertencentes ao conjunto robusto  $\mathbf{X}_d^*$ . No entanto, para examinar com mais detalhes a proximidade com o conjunto solução contabilizou-se também TE com  $\varepsilon_{TE} = 0.01$ . A Fig. 4.38 (gráfico superior) apresenta que as populações maiores contêm mais menores taxas de erro independentemente da geração pesquisada. O gráfico inferior apresenta que o número de indivíduos

iguais no conjunto solução obedece uma tendência independente relacionada ao tamanho da população.

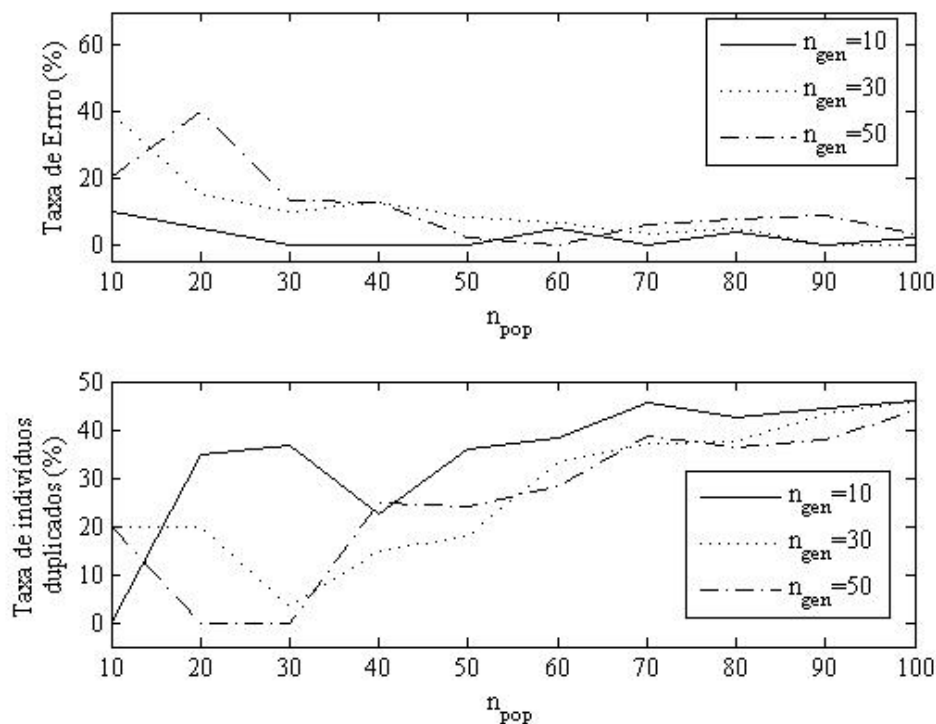


Fig. 4.38: SCH - [I]RMOEA: métrica TE. Em (a), a métrica TE utilizando  $\varepsilon_{TE} = 0.01$ . Em (b), o número percentual de soluções iguais dentro de  $\mathbf{X}_s^*$ . Pelas informações de (a) e (b) pode-se estimar o número de soluções diferentes produzidas pelo [I]RMOEA.

A uniformidade do conjunto solução foi medida pela métrica MB[I]; o resultado encontra-se plotado na Fig. 4.39. A aproximação do valor normalizado  $MB[I]_{norm}$  ao valor zero indicou melhoria da uniformidade do espaçamento à medida que a população cresceu. Em outras palavras, percebeu-se que populações maiores atingem os melhores índices  $MB[I]_{norm}$ .

A imagem da fronteira robusta está plotada na Fig. 4.40. Como esperado, a simulação com população maior supera em qualidade a outra simulação.

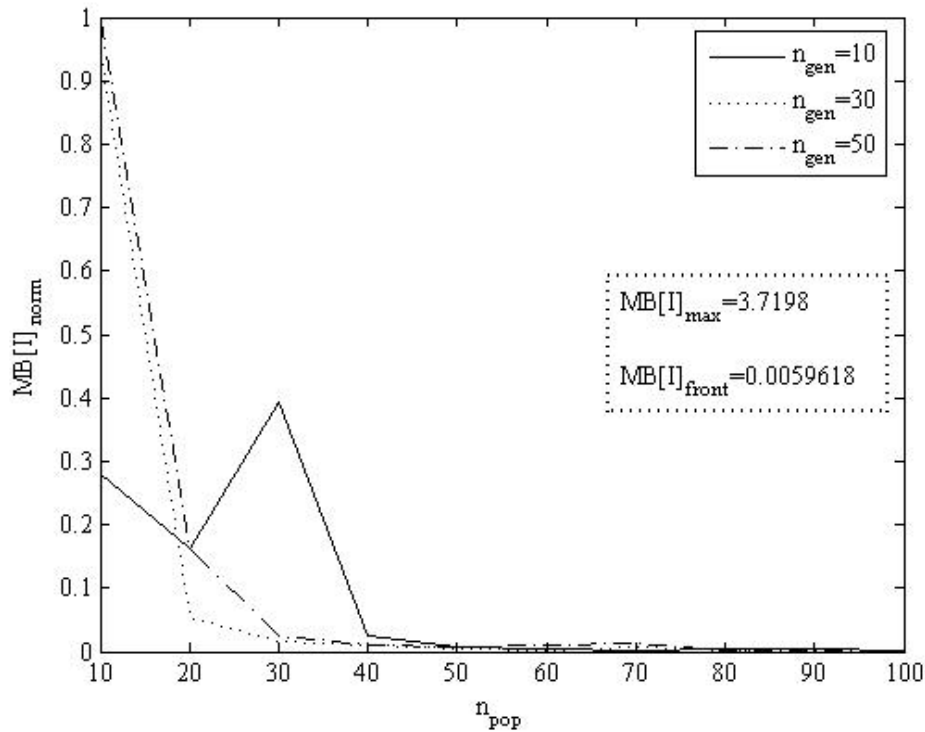
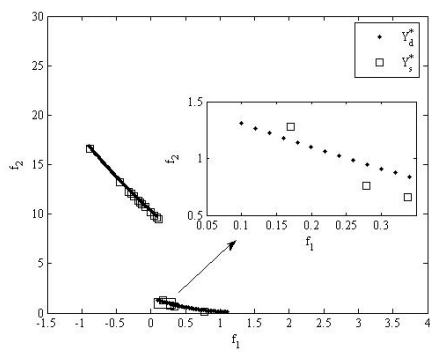
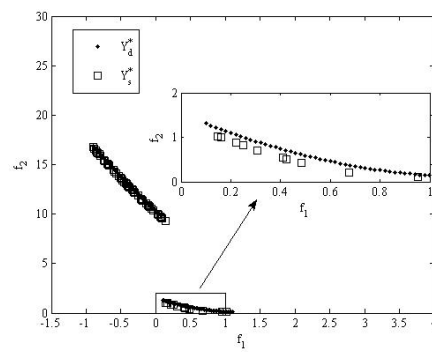


Fig. 4.39: SCH - [I]RMOEA: métrica  $MB[I]$ . A partir de  $n_{pop} = 40$ , a métrica indica ganho em uniformidade para todas as simulações.



(a) SCH2 ( $n_{pop} = 50$ ;  $n_{gen} = 20$ ).



(b) SCH2 ( $n_{pop} = 80$ ;  $n_{gen} = 100$ ).

Fig. 4.40: SCH2 - [I]RMOEA: paralelo entre  $Y_d^*$  e  $Y_s^*$ . No gráficos em detalhe, é mostrada a proximidade e distribuição das populações ao longo da fronteira robusta.

[I]RMOEA  $\times$  FON

Para a função FON, utilizou-se  $n_{pop} = [10 : 10 : 100]$  e  $n_{gen} = [10 : 10 : 50]$ . A convergência do [I]RMOEA para a função é mais lenta que para a função SCH2, como pode ser visto na Fig. 4.41. Um dos fatores que retardaram a convergência foi o aumento do número de variáveis de busca. Na próxima seção, esse ponto ficará ainda mais claro com os testes realizados com mais variáveis. Novamente, nota-se que o tamanho da população é um fator determinante na convergência do algoritmo.

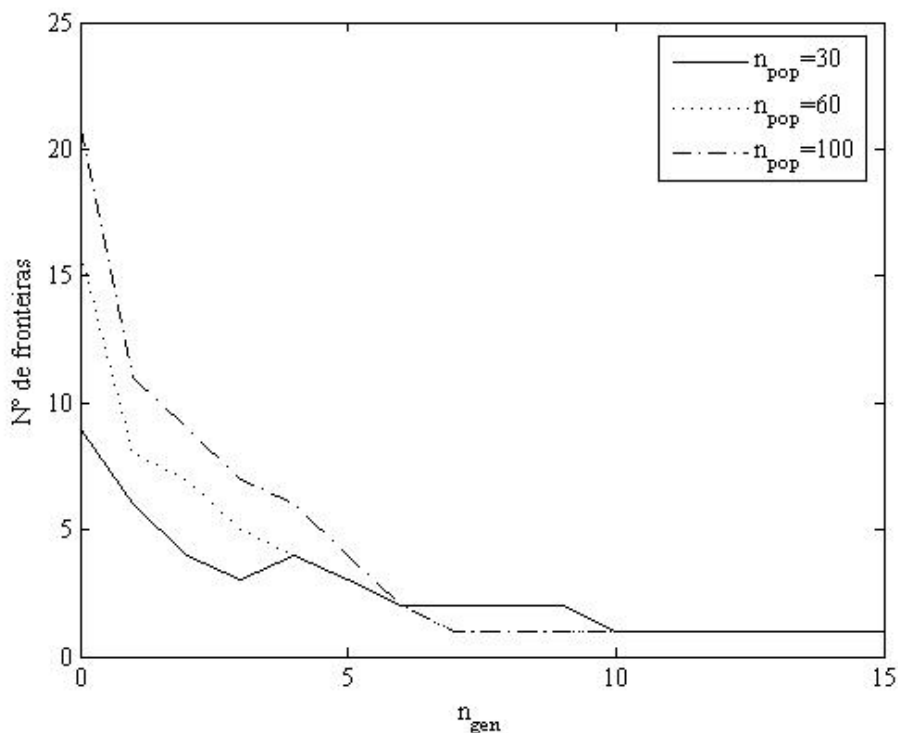


Fig. 4.41: FON - [I]RMOEA: Convergência das fronteiras. O número de fronteiras decresce mais lentamente que para a função SCH2.

A eficiência do algoritmo frente à função está mostrada na Fig. 4.42. A TE foi utilizada com precisão de  $\varepsilon_{TE} = 0.05$ . Observe que a métrica resultou em valores relativamente altos para as populações menores ( $n_{pop} = 10$ ); e valores baixos a partir de  $n_{pop} = 50$ . No gráfico inferior, a simulação com  $n_{pop} = 10$  apresentou uma evolução instável do número de indivíduos duplicados. De fato, as populações menores (pequenas amostras) são mais sensíveis à ação dos operadores evolucionários.

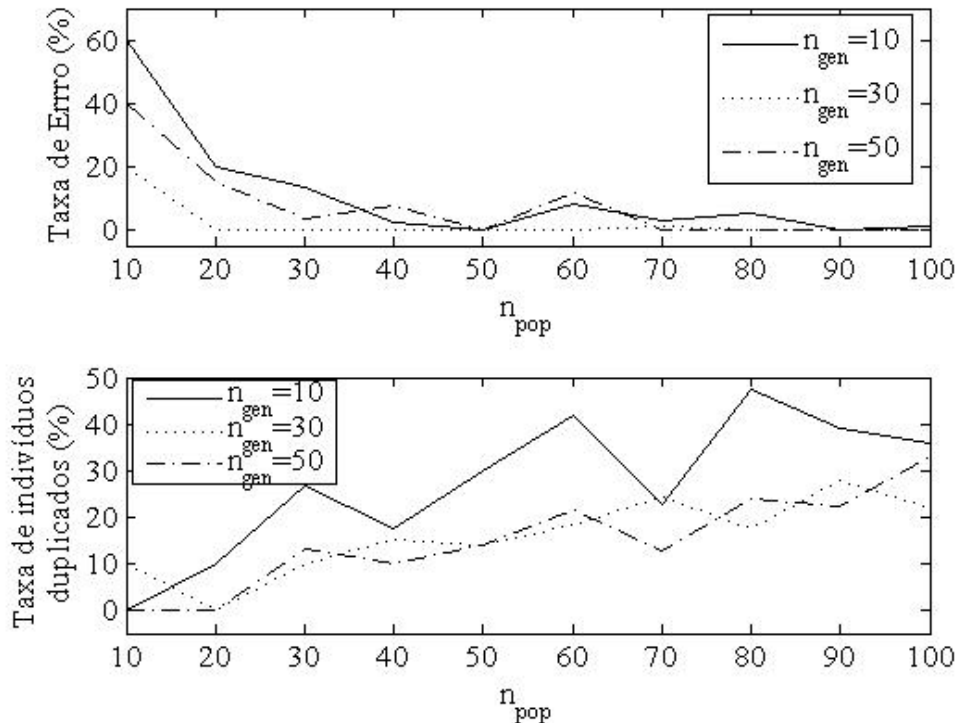


Fig. 4.42: FON - [I]RMOEA: métrica TE. Em (a), a métrica TE utilizando  $\varepsilon_{TE} = 0.05$ . Em (b), o percentual de soluções iguais dentro de  $\mathbf{X}_s^*$ .

O resultado da avaliação da métrica MB[I] para a função FON está na Fig. 4.43. A uniformidade no espaço dos objetivos nesta função foi relativamente difícil de ser obtida quando comparada às outras funções teste. Neste caso, mesmo se a população estivesse uniformemente distribuída no espaço de busca, ela se apresentaria mais concentrada no espaço dos objetivos. Essa característica da função FON é devida ao termo exponencial em sua expressão formal. Novamente, as populações maiores atingem os melhores índices de uniformidade.

Duas imagens robustas foram escolhidas para exemplificar a saída do [I]RMOEA, veja-as na Fig. 4.44. A simulação com população de 100 indivíduos apresenta todos os pontos na fronteira robusta e com distância visualmente homogênea. Essa informação está em concordância com os gráficos da TE e MB[I] apresentados anteriormente.

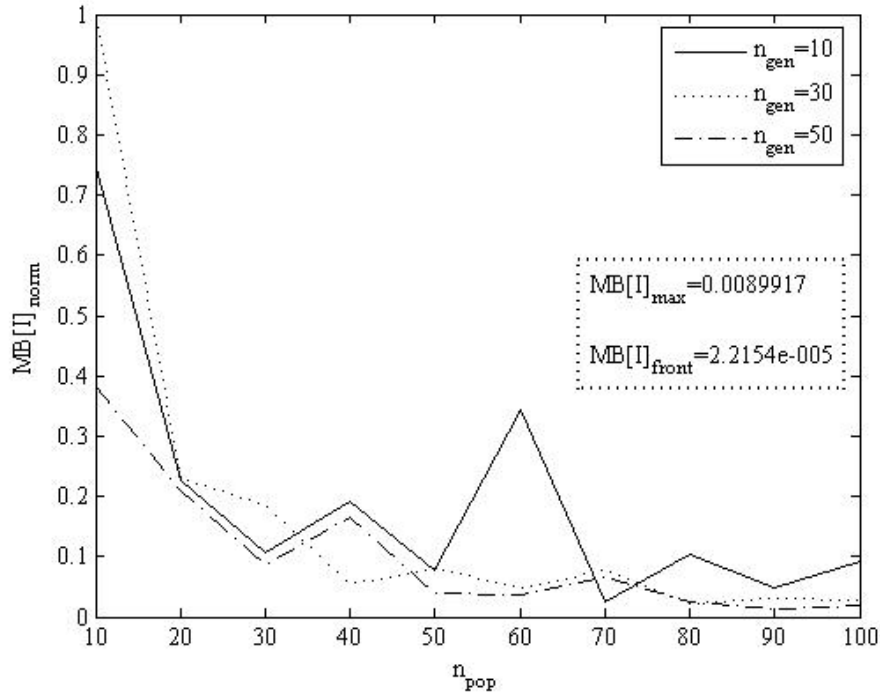
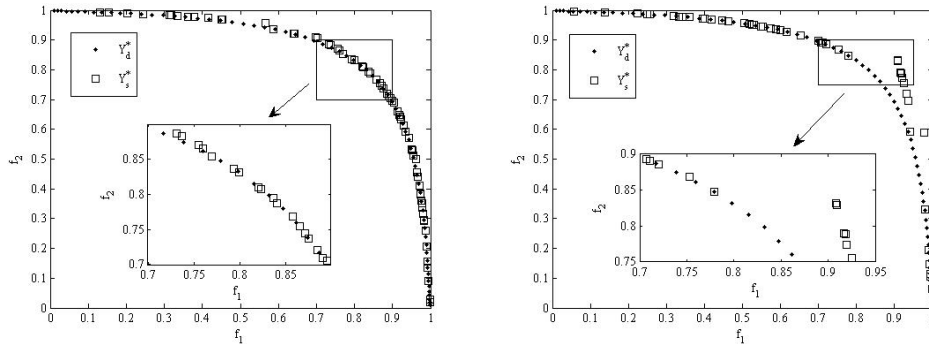


Fig. 4.43: FON - [I]RMOEA: métrica  $MB[I]$ . A uniformidade e convergência mais suave ocorreu para populações maiores.



(a) FON ( $n_{pop} = 100$ ;  $n_{gen} = 50$ ).

(b) FON ( $n_{pop} = 60$ ;  $n_{gen} = 50$ ).

Fig. 4.44: FON - [I]RMOEA: paralelo entre  $Y_d^*$  e  $Y_s^*$ . Em (a), as fronteiras  $Y_d^*$  e  $Y_s^*$  confundem, devido à proximidade. Em (b), percebe-se a presença de alguns pontos fora da fronteira e sem espaçamento uniforme.

$$[I]RMOEA \times ZDT1$$

Para a função ZDT1, utilizou-se  $n_{pop} = [10 : 10 : 100]$  e  $n_{gen} = [10 : 10 : 100]$ .

A convergência do [I]RMOEA em termos de fronteiras é mostrada na Fig. 4.45. A partir geração 13, todas as simulações convergiram para a primeira fronteira. Uma vez que a população encontra-se na primeira fronteira, o processo de seleção e a técnica de nicho buscaram manter um nível aceitável de diversidade. Esta questão é tratada na Fig. 4.46 que mostra número de indivíduos repetidos.

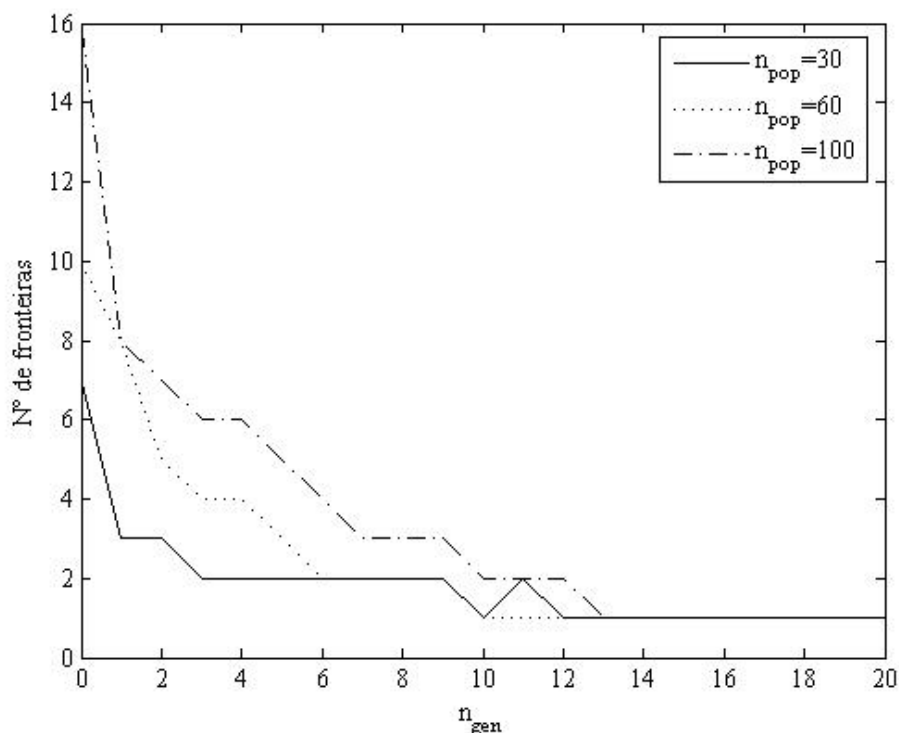


Fig. 4.45: ZDT1 - [I]RMOEA: Convergência das fronteiras. As populações maiores oferecem, inicialmente, mais diversidade em número de fronteiras. Nessas simulações, a diversidade inicial das fronteiras não foi suficiente para impedir a convergência simultânea para primeira fronteira na mesma geração.

A Fig. 4.46 apresenta os resultados da métrica TE com precisão de  $\varepsilon_{TE} = 0.01$ . A métrica apresentou, a partir de  $n_{pop} = 50$ , que o [I]RMOEA produz conjuntos solução com todos os elementos do conjunto na fronteira robusta. Além disso, no gráfico inferior, o número de indivíduos duplicados é mais baixo que aqueles apresentados para outras funções teste. Isso ocorre pelo motivo de ZDT1 conter fronteira convexa e contínua, facilitando exploração de regiões robustas.

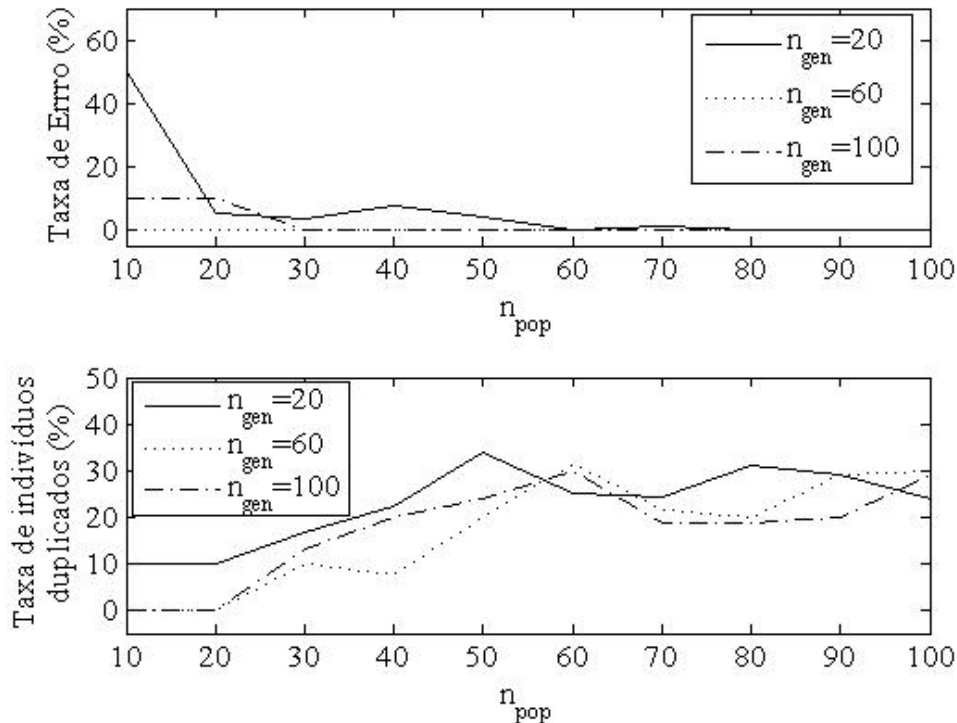


Fig. 4.46: ZDT1 - [I]RMOEA: métrica TE. Acima, a TE indica alto nível de conformidade para populações maiores. Abaixo, o número de indivíduos iguais foi o mais baixo dos testes realizados.

O nível de uniformidade de espaçamento entre os elementos dos conjuntos solução está apresentado na Fig. 4.47. Uma vez que as simulações mostradas na Fig. 4.45 apresentaram a convergência para primeira fronteira ocorreu na 13ª geração, decidiu-se por examinar como a uniformidade se modifica para gerações maiores que 20. Por isso, a métrica MB[I] foi utilizada para gerações superiores a 20. Concluiu-se, que mesmo após a convergência a primeira fronteira, o algoritmo necessita de algumas gerações para atingir níveis aceitáveis de homogeneidade.

A Fig. 4.48 mostra as fronteiras robustas obtidas para duas execuções do [I]RMOEA. Há regiões com melhor representatividade de soluções que outras, principalmente para o caso (a). No entanto, considerando a natureza estocástica dos algoritmos evolucionários e o baixo índice de duplicatas, avaliou-se positivamente a qualidade de distribuição das soluções na fronteira robusta.

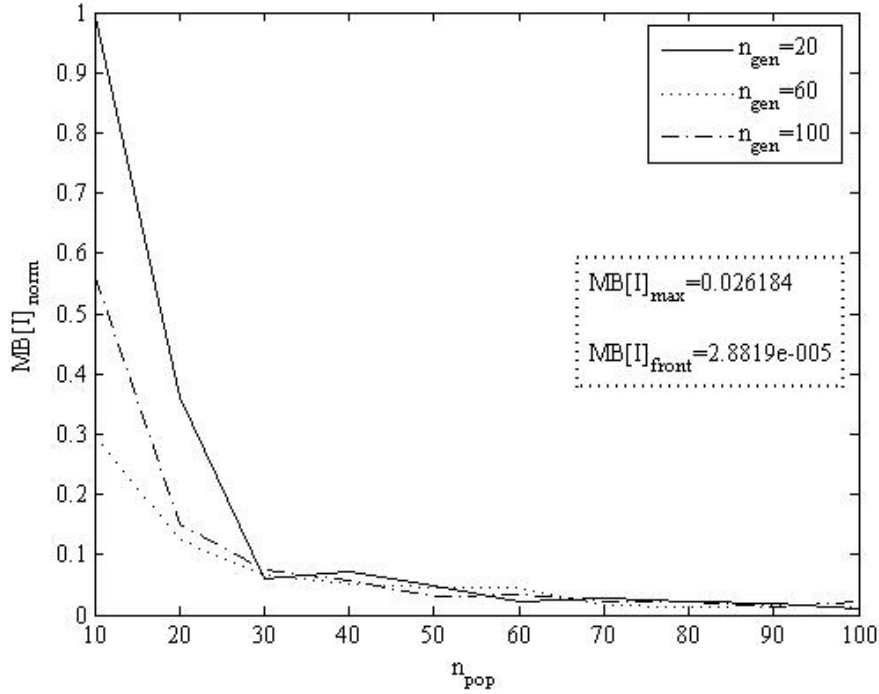
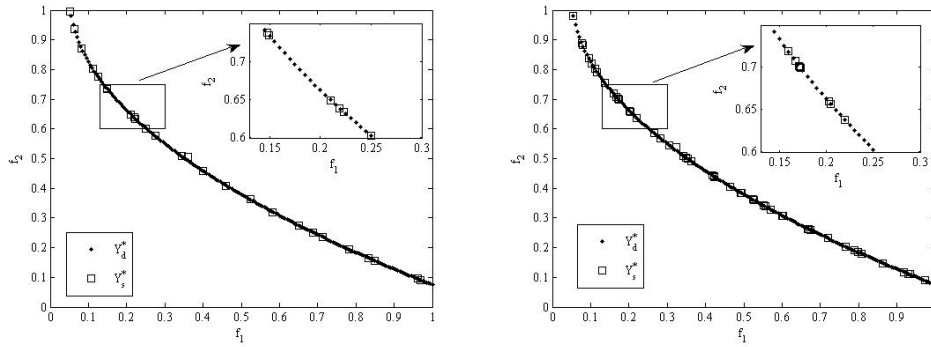


Fig. 4.47: ZDT1 - [I]RMOEA: métrica MB[I]. A métrica indica que a melhoria da uniformidade depende mais de  $n_{pop}$  que de  $n_{gen}$ .



(a) ZDT1 ( $n_{pop} = 30$ ;  $n_{gen} = 60$ ).

(b) ZDT1 ( $n_{pop} = 70$ ;  $n_{gen} = 80$ ).

Fig. 4.48: ZDT1 - [I]RMOEA: paralelo entre  $\mathbf{Y}_d^*$  e  $\mathbf{Y}_s^*$ . Ambos gráficos confirmam a eficiência em encontrar a fronteira de Pareto robusta e a qualidade de distribuição das soluções sobre ela.

$$[I]RMOEA \times ZDT2$$

Para a função ZDT2, utilizou-se  $n_{pop} = [10 : 10 : 100]$  e  $n_{gen} = [10 : 10 : 100]$ .

Alguns testes foram realizados até a 100<sup>a</sup> geração. No entanto, em todas as simulações a convergência para a primeira fronteira ocorreu antes da 20<sup>a</sup> geração. O Resultado da convergência é mostrado na Fig. 4.49. A função ZDT2 é não convexa e restrita. Observou-se que tratamento de restrições do [I]RMOEA não influenciou em sua convergência das fronteiras porque é feito independentemente de outros processos internos do algoritmo. Quanto a não convexidade, os testes não foram conclusivos para compreender a relação com a convergência de fronteiras. Entretanto, comparando-se ZDT1 e ZDT2, ZDT2 teve convergência mais lenta.

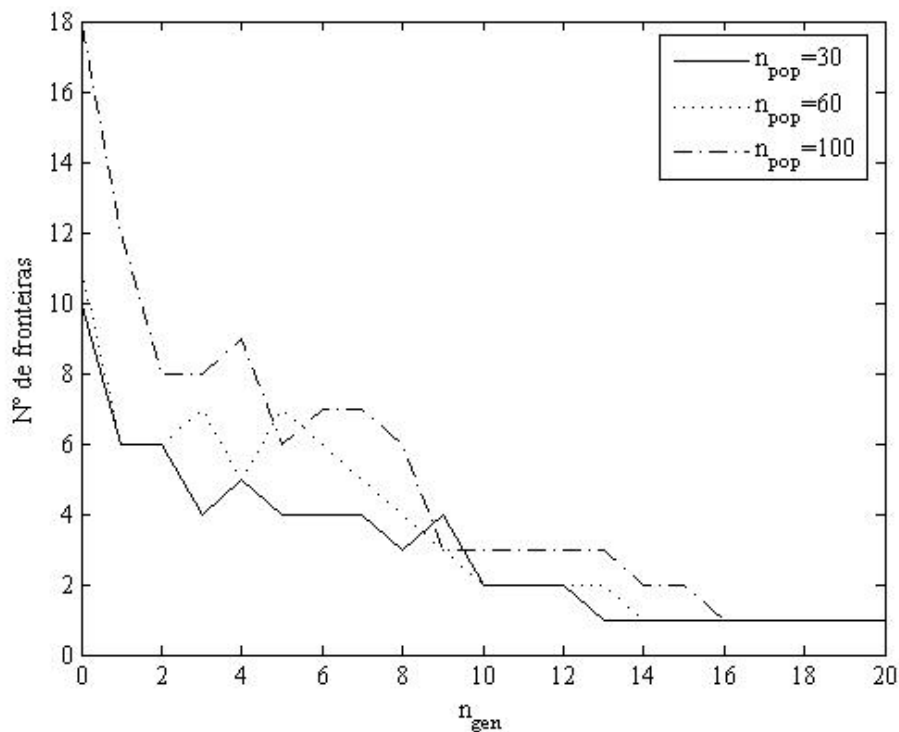


Fig. 4.49: ZDT2 - [I]RMOEA: Convergência das fronteiras. A presença de restrições não interfere na convergência de fronteiras. Por outro lado, populações maiores oferecem mais diversidade inicialmente e retardam, mesmo que suavemente, a convergência do algoritmo.

A Fig. 4.50 apresenta os resultados da métrica TE com precisão de  $\varepsilon_{TE} = 0.01$ . A métrica apresentou melhor conformidade dos resultados para simulação com  $n_{pop} = 100$ . Isso significa que o [I]RMOEA produziu conjuntos solução completamente situados na fronteira robusta. No gráfico inferior, o número de indivíduos duplicados manteve-se abaixo dos 40%. Resultado

considerado normal frente ao testes já apresentados.

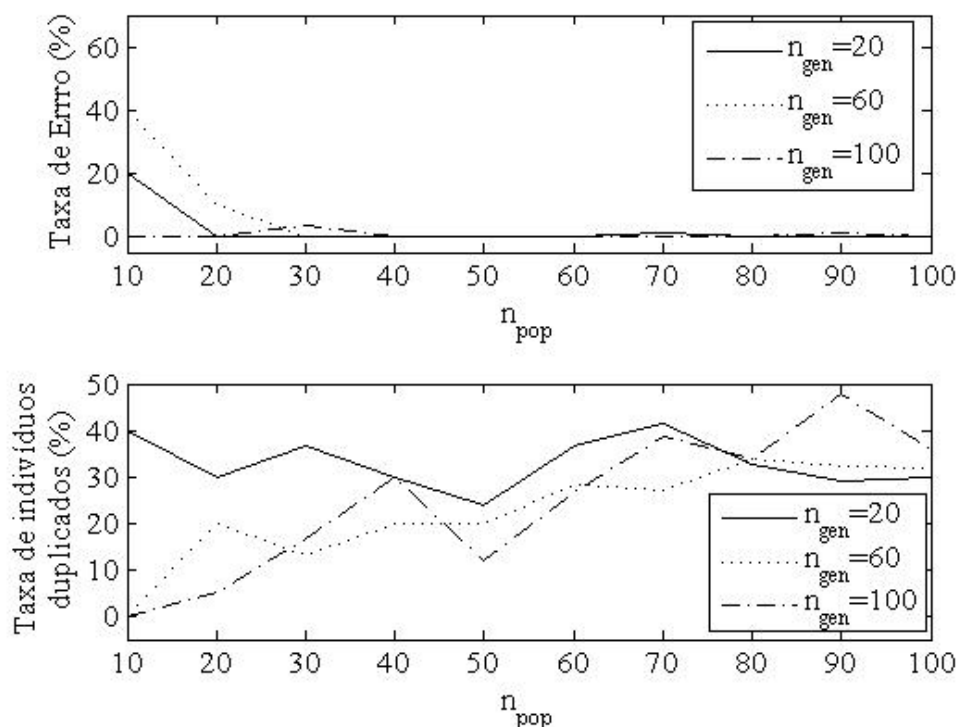


Fig. 4.50: ZDT2 - [I]RMOEA: métrica TE. Em (a), [I]RMOEA provou ser eficiente para resolver o problema ZDT2. Em (b), o percentual de soluções iguais dentro de  $\mathbf{X}_s^*$  permanece com valor abaixo de 40%.

O resultado da métrica MB[I] está apresentado na Fig. 4.51. Observa-se que em pelo menos duas simulações ( $n_{pop} = 70$ ;  $n_{gen} = 60$ ) e ( $n_{pop} = 90$ ;  $n_{gen} = 100$ ) a métrica avaliou erroneamente a distribuição dos pontos em  $\mathbf{X}_s^*$  melhor que em  $\mathbf{X}_d^*$ . Essa falha ocorreu porque esses conjuntos  $\mathbf{X}_s^*$  geraram um ponto próximo à região não viável entre as partes descontínuas da fronteira robusta, causando uma melhoria “falsa” de distribuição dos pontos. Esse ponto pode ser visto na Fig. 4.52.

Uma função de inclusão foi empregada para avaliar a viabilidade de pontos solução gerados, na inicialização do algoritmo e após a troca de material genético através do cruzamento. A Fig. 4.52 mostra duas simulações onde pode ser notado a ausência de pontos na região da função de restrição.

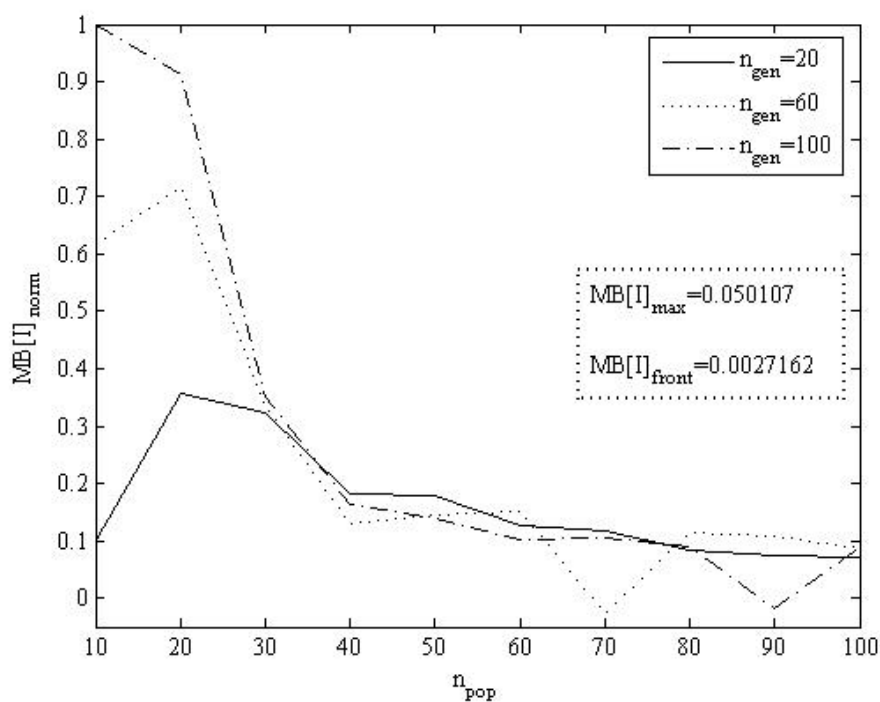
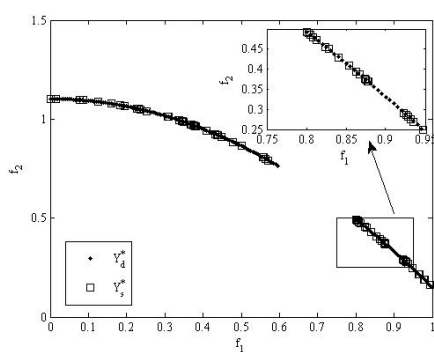
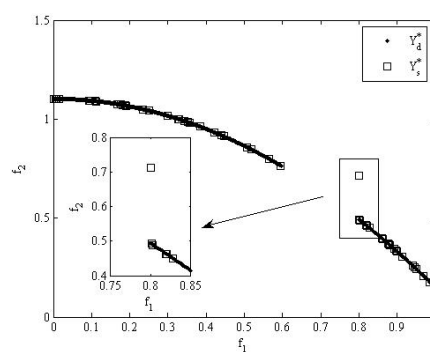


Fig. 4.51: ZDT2 - [I]RMOEA: métrica  $MB[I]$ . A métrica falha nos pontos abaixo de  $MB[I]_{norm} = 0$ , devido à presença de falsa solução robusta próxima à região não viável.



(a) ZDT2 ( $n_{pop} = 100$ ;  $n_{gen} = 100$ ).



(b) ZDT2 ( $n_{pop} = 70$ ;  $n_{gen} = 60$ ).

Fig. 4.52: ZDT2 - [I]RMOA II: paralelo entre  $\mathbf{Y}_d^*$  e  $\mathbf{Y}_s^*$ . Em (a), a imagem da região não viável não contém pontos. Em (b), a solução isolada em  $\mathbf{Y}_s^*$  é, de fato, ineficiente.

[I]RMOEA  $\times$  ZDT3

Para a função ZDT3, utilizou-se  $n_{pop} = [10 : 10 : 100]$  e  $n_{gen} = [10 : 10 : 100]$ .

A Fig. 4.53 expõe três curvas de convergência do [I]RMOEA. O decréscimo do número de fronteiras é esperado conforme já visto nas simulações anteriores. No entanto, a simulação com  $n_{pop} = 60$  indivíduos apresentou um traçado interessante; em três gerações o número de fronteiras cresceu. Isto ocorreu quando o algoritmo encontra um ou mais pontos em uma região robusta diferente. Na verdade, desde o início das gerações o [I]RMOEA encontra novas regiões com robustez melhor que as anteriores. Entretanto, pelo gráfico apresentado só é possível visualizar quando o número de fronteiras cresce. Após encontrar uma nova região, o [I]RMOEA busca convergir toda a população para a primeira fronteira.

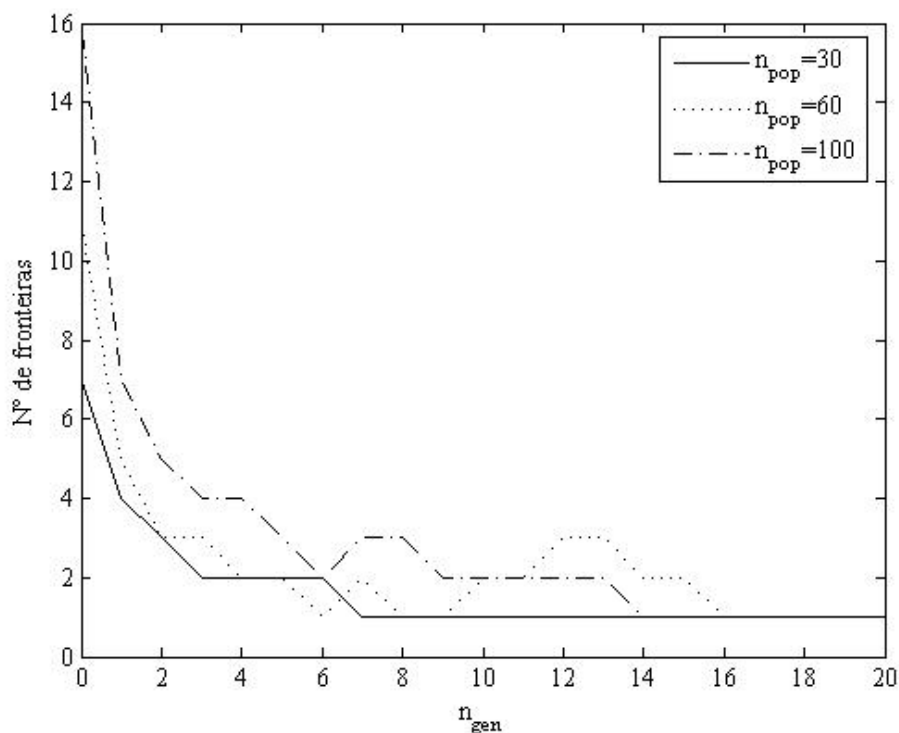


Fig. 4.53: ZDT3 - [I]RMOEA: Convergência das fronteiras. As gerações onde há crescimento do número de fronteiras indicam descoberta de novas regiões robustas.

A Fig. 4.54 apresenta os resultados da métrica TE com precisão de  $\varepsilon_{TE} = 0.01$ . Conforme anteriormente, a métrica comprovou que os melhores resultados ocorrem para populações maiores. A partir de  $n_{pop} = 50$ , o [I]RMOEA pro-

duz conjuntos solução com grande parte de seus os elementos do conjunto na fronteira robusta. O gráfico inferior mostra que número de indivíduos iguais no conjunto solução é próximo daquele contabilizado para as demais funções apresentadas da família ZDT. Dentre todas as funções, ZDT3 foi considerada a mais difícil de se encontrar o conjunto solução robusto, talvez por causa da continuidade de sua fronteira robusta ser interrompida em quatro regiões.

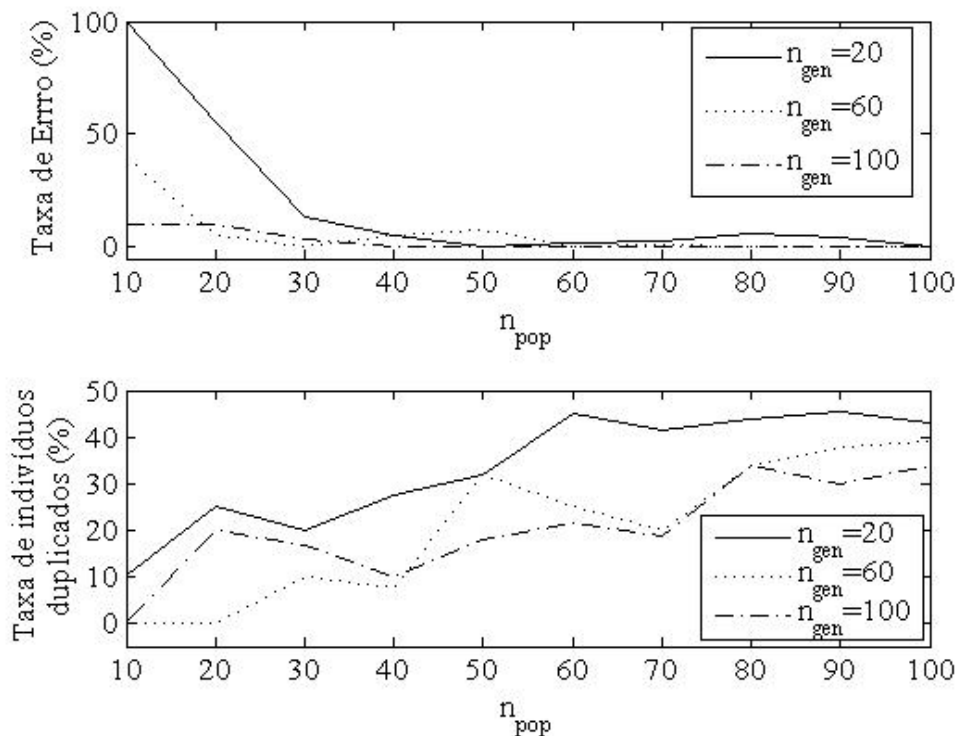


Fig. 4.54: ZDT3 - [I]RMOEA: métrica TE. Em (a), a TE utilizando  $\varepsilon_{TE} = 0.01$ ; em (b), o percentual de soluções iguais dentro de  $\mathbf{X}_s^*$ .

A uniformidade do conjunto solução avaliada pela métrica MB[I] está apresentada na Fig. 4.55. A homogeneidade das soluções robustas para a função ZDT3 foram consideradas as piores de todas as funções teste investigadas. No entanto, a figura mostra que indicadores equivalentes às demais simulações foram obtidos para as populações maiores, neste caso  $n_{pop} = 100$ .

Duas imagens robustas foram escolhidas para exemplificar a saída do [I]RMOEA frente ZDT3, veja-as na Fig. 4.56. Ambas simulações contêm todos os indivíduos locados na fronteira robusta. Contudo, a distribuição da população com 80 indivíduos é visualmente mais uniforme que no outro caso.

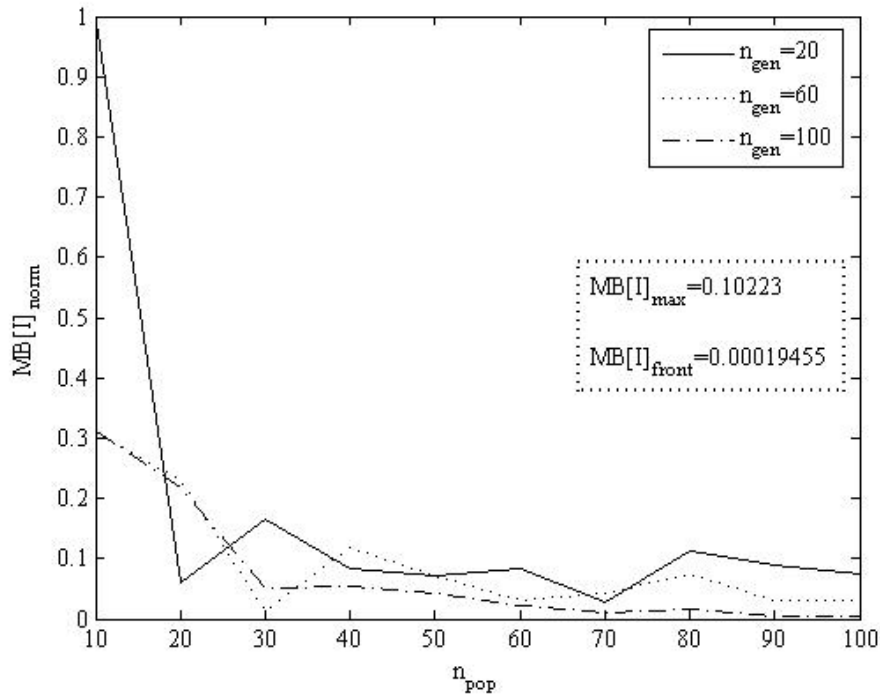
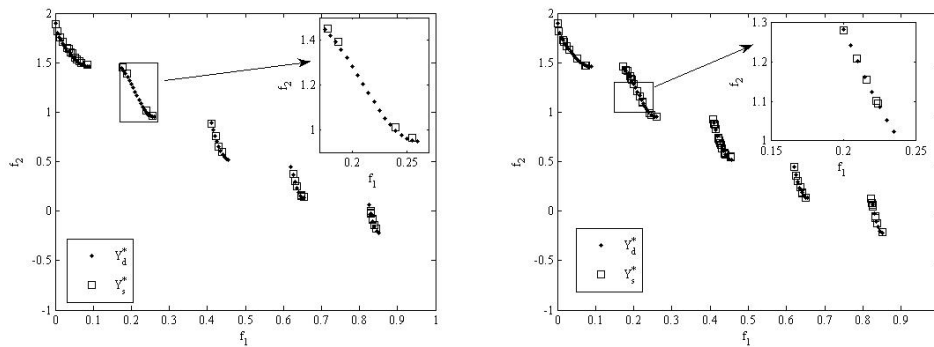


Fig. 4.55: ZDT3 - [I]RMOEA: métrica  $MB[I]$ . A métrica aponta dificuldade de se obter uniformidade no conjunto solução para  $n_{pop}=20$  e  $n_{pop}=60$ .



(a) ZDT3 ( $n_{pop} = 50$ ;  $n_{gen} = 20$ ).

(b) ZDT3 ( $n_{pop} = 80$ ;  $n_{gen} = 100$ ).

Fig. 4.56: ZDT3 - [I]RMOA II: paralelo entre  $Y_d^*$  e  $Y_s^*$ . Em (a), pode se notar concentrações de soluções de forma não homogênea, como no gráfico em detalhe. Em (b), a imagem robusta está uniformizada.

### Considerações sobre o [I]RMOEA

Levando em conta os resultados obtidos pelas métricas TE e MB[I] ao longo das simulações, considerou-se que o [I]RMOEA obteve sucesso em todos os casos de teste. Entretanto, alguns pontos merecem ainda ser comentados.

**Custo fixo.** Uma grande vantagem em muitos algoritmos evolucionários é manter o esforço computacional ao longo das gerações. Por exemplo, considerando-se que o critério de parada seja por número máximo de gerações, assumindo-se que a população permaneça com mesmo tamanho e os demais processos evolucionários seguirem um padrão, então o custo computacional total pode ser estimado anteriormente à execução do algoritmo.

**Critério de parada.** O término do processo evolutivo foi estabelecido pela escolha, *a priori*, de um valor máximo para o número de gerações. No entanto, as figuras de convergência, como por exemplo Fig. 4.41 e Fig. 4.45, mostram que a convergência à primeira fronteira ocorreu antes da vigésima geração, isso foi verificado para todas as simulações. Por isso, pelo menos dois outros critérios de parada podem ser inseridos. O primeiro critério estaria relacionado à convergência do algoritmo, ou seja, o [I]RMOEA poderia ser finalizado após um número fixo de gerações que o algoritmo permanecesse com todos os indivíduos apenas na primeira fronteira. O segundo, seria um critério complementar ao descrito anteriormente. A técnica de nicho seria empregada para qualificar a distribuição de pontos na fronteira. O número de rodadas de bisseção seria aumentado em uma ou duas unidades, de forma que mais nichos fossem definidos sobre a fronteira. Como o processo de seleção busca distribuir os pontos para os nichos, espera-se um resultado ainda mais homogêneo na fronteira de Pareto Robusta.

**Convergência.** A convergência para primeira fronteira é desejada. Todavia, uma vez que a primeira fronteira é atingida por toda a população, é necessário manter a diversidade diminuindo ao máximo o número de duplicatas. Duas opções podem ser facilmente implementadas. Inicialmente, deve-se definir um índice de diversidade baseado, por exemplo, no número de indivíduos iguais dentro da primeira fronteira. De acordo com o índice: a) a probabilidade de mutação poderia ser aumentada dinamicamente para inserir diversidade, como por Soares [109]; ou b) o número de rodadas de bisseção seria alterado de forma a proporcionar a criação de mais nichos sobre a fronteira, permitindo ao processo de seleção distribuir melhor a população.

**Tratamento de restrições.** As restrições são escritas em forma de funções de inclusão, desta forma realiza-se a classificação de regiões como viáveis ou não. As regiões não viáveis são excluídas definitivamente e, as regiões declaradas como viáveis suportam o [I]RMOEA na oferta de soluções candidatas aleatórias, quando da criação da população inicial ou para substituir

uma solução não viável fruto de uma operação como o cruzamento ou mutação.

Computação da robustez. Dada uma população de soluções candidatas, o pior caso da presença de incertezas é computado ponto a ponto, empregando-se uma função de inclusão mínima. O algoritmo da Tab.3.8 mostra como é simples o processo definição do valor robusto para os indivíduos da população.

O [I]RMOEA contém muitas vantagens, entre elas destacam-se: a) o custo fixo por geração; b) a computação da robustez utilizando-se de intervalos; c) a exclusão de espaços não viáveis definitivamente; e d) a técnica de nicho simples, eficiente e flexível. Dentre os três algoritmos robustos, o [I]RMOEA se apresenta como o mais promissor para tratar problemas multidimensionais.

#### 4.4 Testes Complementares

Os algoritmos [I]RMOA I e [I]RMOA II que são puramente determinísticos e suportados por intervalos, são sensivelmente dependentes da dimensionalidade dos espaços de busca e de incerteza, espaços onde ocorrem as bissecções. Por isso, a garantia de resolver o problema de otimização custa caro. Por outro lado, sem a vantagem de garantir a convergência para o resultado correto, o método [I]RMOEA é independente da dimensionalidade do espaço de busca, sendo a estrutura de dados de tamanho fixo geração após geração. Contudo, o [I]RMOEA é dependente do espaço de incerteza, pois a computação da interferência das incertezas é similar ao [I]RMOA II. A questão da dimensionalidade é tratada nesta seção para o [I]RMOA II e [I]RMOEA. Para o [I]RMOA I, apresenta-se como ele pode ser utilizado como ferramenta de validação.

##### 4.4.1 [I]RMOA I como Ferramenta de Validação

Como dito anteriormente, o [I]RMOA I é uma ferramenta para validação de outros métodos multi-objetivos robustos. Os próximos experimentos, retirados de Soares *et al* [112], foram dedicados a mostrar como isso pode ser feito. O primeiro passo é descobrir, para dado problema, os valores dos parâmetros  $\varepsilon_{[x]}$  e  $\varepsilon_{x\#}$  que produzem um envelope mínimo sobre a fronteira robusta. A comprovação que os envelopes estão corretos deve ser feita via comparação do conjunto solução encontrado a um conjunto solução *referência*, conhecido previamente para um dado valor de incerteza, por exemplo,  $\mathbf{p}_1$ . Esse também é o momento de testar e tentar encontrar melhores funções de inclusão. Assumindo que o conjunto solução do problema possa variar conforme o valor do parâmetro de incerteza, então o conjunto utilizado como referência para

calibrar o [I]RMOA I perderia a validade quando o valor de incerteza fosse alterado para  $\mathbf{p}_2$ . Por outro lado, o processo do [I]RMOA I continua válido, bastando apenas ser executado uma vez mais para o novo valor de incerteza. Depois disso, o novo envelope pode ser utilizado para validar outros algoritmos.

As funções teste FON e ZDT3 foram as escolhidas para realizar os experimentos. Foram aproveitados alguns resultados das execuções realizadas para descrever a Subseção 4.3.1, que refere aos testes do [I]RMOA I. Veja na Fig.4.57 as configurações que geraram os envelopes com qualidade satisfatória sobre a fronteira robusta. Para simplificar os experimentos, o parâmetro  $\mathbf{p}$

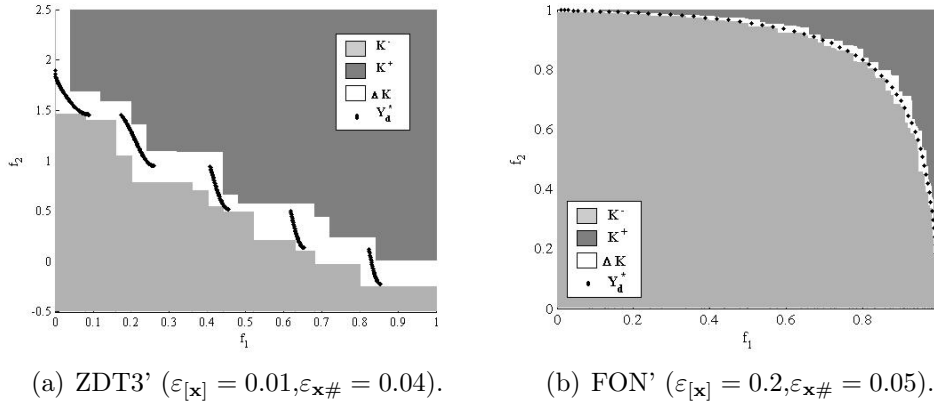


Fig. 4.57: Envelopes sobre ZTD3 e FON (fonte: resultados de testes realizados para Subseção 4.3.1).

que define a posição da fronteira robusta manteve-se constante, e um algoritmo puramente evolucionário foi empregado como o método a ser validado pelo [I]RMOA I. Trata-se de um algoritmo genético multi-objetivo com vetor das funções objetivo adaptado para lidar com problemas de otimização robusta. Doravante esse algoritmo é chamado de RMOGA (Robust Multi-Objective Genetic Algorithm), e possui a seguinte configuração: a) codificação binária; b) cruzamento com um ponto de corte; c) mutação pela mudança de bits; d) seleção pelo método da Roleta (desempenho individual calculado por (3.37)); e) técnica de nicho NB[I] (veja Subseção 3.6.2); e f) elitismo utilizado no NSGA-II [34]. A descrição completa do algoritmo pode ser encontrada em Soares *et al.* [110]. Para esse método evolucionário, o problema multi-objetivo robusto 3.9, considerando a filosofia do pior caso, deve ser redefinido para funcionar segundo a estrutura estocástica, então

$$\min_{\mathbf{x} \in \mathbf{X}} \max_{\mathbf{p}' \in \mathbf{P}'} \mathbf{f}(\mathbf{x}, \mathbf{p}'), \tag{4.12}$$

sendo  $\mathbf{P}'$  um conjunto finito de amostras selecionadas aleatoriamente de  $\mathbf{P}$ . Considerando que a qualidade da amostragem aumenta com a quantidade de elementos coletados, então quanto maior o número de amostras mais o algoritmo se aproxima da fronteira robusta correta. Todavia, esse aumento da qualidade acarreta um indesejável crescimento do tempo consumido no processo de otimização. Assim, para tentar encontrar uma relação custo-benefício vantajosa à aplicação do algoritmo estocástico, o [I]RMOGA I foi empregado para validar um tamanho de amostra  $\mathbf{P}'$  tal que o algoritmo estocástico gere a fronteira robusta correta.

Nos testes de validação, os valores dos parâmetros do RMOGA foram fixados em  $p_c = 1$ ,  $p_m = 0.0005$ ,  $n_{pop} = 50$  e  $n_{gen} = 40$ . Dez diferentes tamanhos de amostra de  $\mathbf{P}'$  foram testados, 5, 10, ..., 50. Em cada teste o RMOGA foi executado 10 vezes, sendo os resultados julgados mais interessantes apresentados na Fig.4.58 e na Fig.4.59.

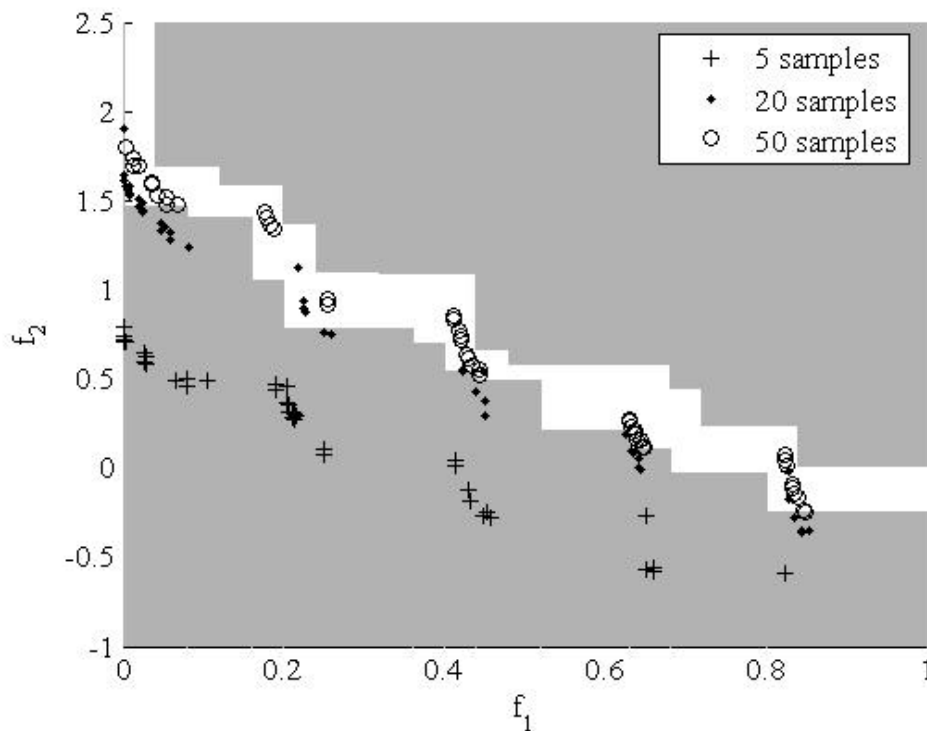


Fig. 4.58: ZDT3 ( $\varepsilon_{[x]} = 0.02, \varepsilon_{x\#} = 0.02$ ) - Validação do RMOGA. As pequenas amostragens produzem piores representações da fronteira de Pareto robusta.

Observe que cada tamanho de amostra de  $\mathbf{P}'$  está associado a uma fron-

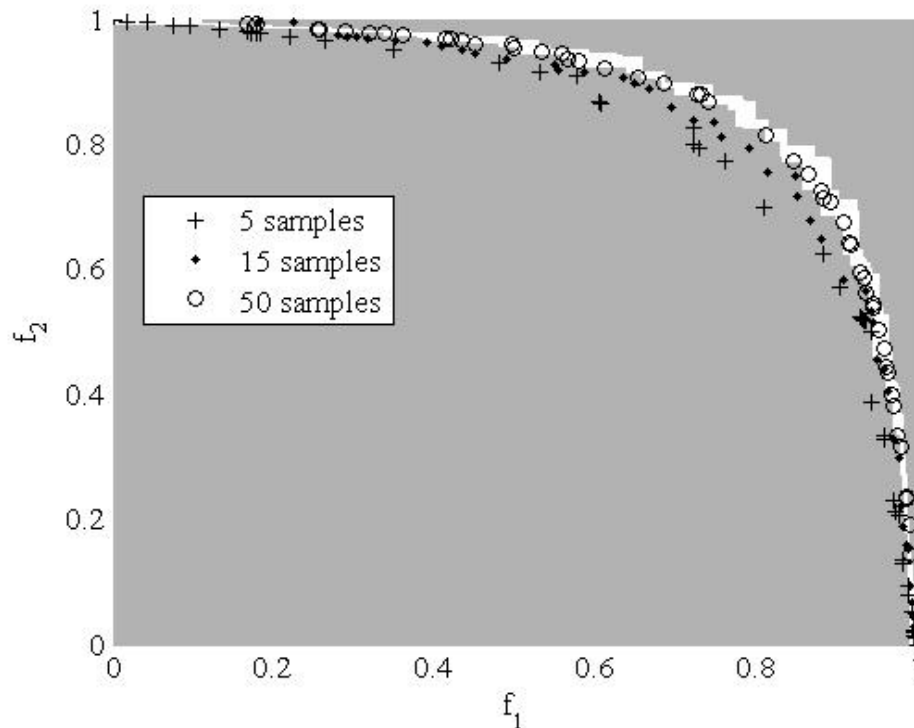


Fig. 4.59: FON ( $\varepsilon_{[x]} = 0.01, \varepsilon_{x\#} = 0.04$ ) - Validação do RMOGA. As pequenas amostragens produzem piores representações da fronteira de Pareto robusta.

teira de Pareto robusta. Um dado tamanho de amostra foi considerado aceitável se a imagem de todo seu conjunto solução ficou situada dentro do envelope gerado pelo [I]RMOA I. Na Fig.4.58 e na Fig.4.59 nota-se que 50 amostras foi tamanho mínimo aceitável. Em outras palavras, para o RMOGA poder ser utilizado na resolução dos problemas robustos ZDT3 e FON é necessário que a amostragem mínima do espaço de incertezas seja de 50 elementos.

#### 4.4.2 [I]RMOA II $\times$ Dimensionalidade do Espaço de Busca

Para esse experimento, somente a versão robusta da função ZDT3 foi utilizada. A precisão para computação da interferência das incertezas foi de  $\varepsilon_{[p]} = 0.01$ . O espaço de busca investigado foi formado por 3 e 4 variáveis, sendo utilizada a precisão  $\varepsilon_{[x]} = 0.04$  para limitar as bissecções no processo de busca. O [I]RMOA II é um método garantido, então os resultados são semelhantes aos apresentados na Subseção 4.3.2, e por isso foram omitidos.

As simulações com número de variáveis acima de 4, para os mesmos valores de  $\varepsilon_{[x]}$  e  $\varepsilon_{[p]}$  citados acima, foram declaradas como “inviáveis” para o equipamento utilizado (CPU Core Duo 1.73GHz/1.73GHz; RAM 1024).

#### 4.4.3 [I]RMOEA × Dimensionalidade do Espaço de Busca

Para essa bateria de testes, somente a versão robusta da função ZDT3 foi utilizada. O espaço de busca investigado foi formado por 5, 10, ..., 30 variáveis, e o número de integrantes da população variou irregularmente de 15 a 300 indivíduos, dependendo do caso. A precisão para computação das incertezas foi  $\varepsilon_{[p]} = 0.01$ . Foram utilizados dois critérios de parada: a) pelo número máximo de 200 gerações; e b) pela estagnação de toda a população na primeira fronteira por 20 gerações consecutivas. De fato, somente as populações com  $n_{pop} \leq 100$  convergiram pelo critério de parada (b). Três testes foram executados em cada caso. Os resultados julgados mais interessantes estão apresentados a seguir.

A Fig.4.60 e Fig.4.61 exemplificam resultados das fronteiras de Pareto robusta para os casos com 10 e 30 variáveis respectivamente. Observe que a proximidade de  $\mathbf{Y}_d^*$  depende do tamanho da população. Analisando o caso mais crítico, ZDT3 com  $n=30$  variáveis, considerou-se o resultado da simulação do [I]RMOEA como satisfatório e seu desempenho ficou comprovado não depender da dimensionalidade no espaço de busca. A Tab.4.61 resume alguns dos resultados obtidos. A métrica TE foi calculada com precisão 0.01 para cada variável, ou seja  $\varepsilon_{TE} = \sqrt{\sum_{i=1}^n 0.01^2}$ , sendo  $n$  o número de variáveis. Similarmente, foram consideradas soluções duplicatas aquelas cujo desvio por variável foi menor que 0.0001. De fato, se for aumentado a probabilidade de

Tab. 4.3: [I]RMOEA - Sumário dos testes complementares.

n	$n_{pop}$	TE (%)	Duplicatas (%)	MB[I]
5	100	1.00	67.0	0.0040
10	100	3.00	52.0	0.0079
15	100	22.0	60.0	0.0244
20	200	1.50	62.5	0.0044
25	250	2.00	56.0	0.0040
30	300	3.33	47.7	0.0050

mutação  $p_m$ , não haveria convergência por estagnação na primeira fronteira. Contudo, a busca poderia se tornar aleatória. Confirmou-se, com esse problema teste, que a diversidade deve vir da amostra (população) e diminuir gradualmente até um ponto de equilíbrio, balanceado pela técnica de nicho.

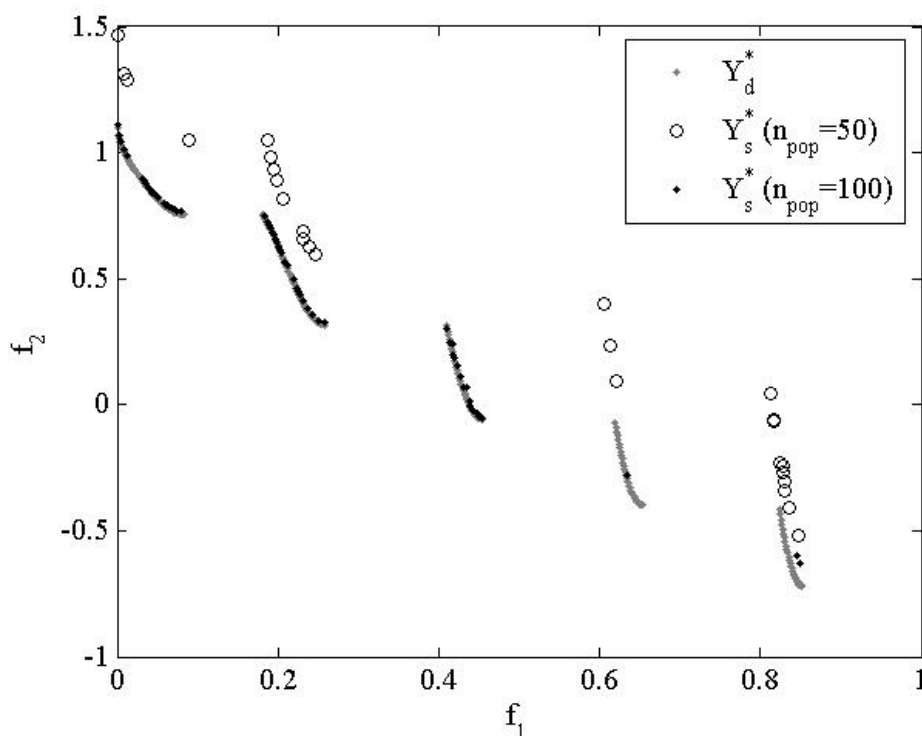


Fig. 4.60: ZDT3 (n=10 variáveis)  $\times$  [I]RMOEA.

Registra-se que pontos na fronteira de Pareto robusta foram obtidos com populações menores. No entanto, quando isso ocorreu a população apresentou índices maiores de duplicatas.

#### 4.5 Considerações Finais

Vários outros testes poderiam ser acrescentados neste capítulo de experimentos. Por exemplo, em se tratando de algoritmos evolucionários uma prática comum tem sido realizar numerosas simulações para verificar a contribuição de cada operador evolucionário. Examinando estas simulações, encontram-se muitas vezes resultados não conclusivos a respeito do desempenho de algoritmos, pois são muitos parâmetros a testar e a relação deles com os problemas teste é certamente particular; cada caso é um caso. Optou-se aqui por concentrar a atenção nos pontos julgados como mais importantes. Os algoritmos puramente intervalares possuem poucos parâmetros e portanto considerou-se fácil a tarefa de testar e demonstrar a dependência de desempenho e de convergência com os parâmetros de precisão tanto no [I]RMOA I quanto no

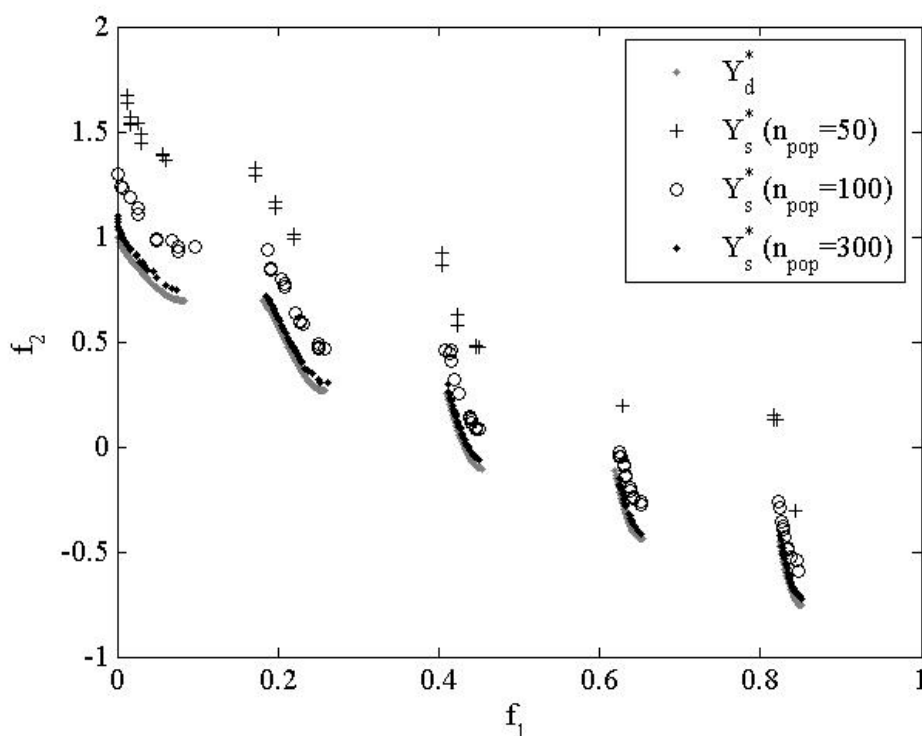


Fig. 4.61: ZDT3 (n=30 variáveis)  $\times$  [I]RMOEA.

[I]RMOA II. Por outro lado, conforme dito anteriormente o [I]RMOEA tem muitos parâmetros. Por isso não foram investigados os diversos operadores e métodos genéticos. Os experimentos cobriram, no entanto, a proximidade da fronteira robusta encontrada com a fronteira robusta computada quando observados o tamanho da população e o número de gerações do processo evolutivo. Os resultados numéricos obtidos foram considerados satisfatórios para a aplicação dos algoritmos desenvolvidos nesse estudo.

Como qualquer grupo de algoritmos de otimização, os algoritmos que utilizam intervalos têm vantagens e desvantagens. Como pontos positivos destacaram-se a garantia de convergência e da computação segura da incerteza, o número reduzido de parâmetros para ajustar, e a simplicidade do código, entre outros. Como aspectos negativos, sobressaiu-se o custo computacional pago por essas vantagens. De qualquer forma, classificou-se os resultados obtidos como “encorajadores”, de modo especial para aqueles problemas robustos que não dependem de resposta imediata. No entanto se há necessidade de considerar questões de memória e/ou velocidade de execução, então é necessário alterar aspectos de precisão e/ou perder a garantia de com-

putação do pior caso de incerteza. Alternativamente, existem outras técnicas a serem inseridas nos algoritmos para acelerar a busca como o uso de derivadas para intervalos ou de técnicas de *propagação intervalar de restrições*, por exemplo. Acelerar o processo de busca é desejável para qualquer método de otimização. Portanto, trata-se de um tópico interessante e motivador para continuar a pesquisa desenvolvida neste trabalho.

## 5. OTIMIZAÇÃO ROBUSTA MULTI-OBJETIVO PARA SINTONIA DE CONTROLADORES PID

A validação dos algoritmos robustos propostos foi realizada considerando-se funções objetivo analíticas cujo conjunto solução pudesse ser reconhecido via análise da expressão formal que lhes definia. As funções teste robustas abrangeram vários tipos de situações normalmente encontradas em situações reais, como descontinuidade e não convexidade da fronteira Pareto robusta. Ainda na validação, foi reforçada a necessidade de se criar funções de inclusão monotônicas, convergentes e mais estreitas possível. Estes quesitos foram atendidos para funções teste, exceto para ZDT2 e ZDT3 que contêm ocorrências múltiplas de variáveis. Normalmente, o desenvolvimento de funções de inclusão parte da expressão analítica que descreve o problema. Como as funções teste continham apenas operadores matemáticos elementares, a função de inclusão natural foi suficiente para a finalidade de teste a que foram propostas. Em problemas reais, a obtenção de funções de inclusão pode ser tarefa complicada. Assim, para verificar que a aplicação dos algoritmos desenvolvidos nesta tese é dependente da modelagem do sistema de otimização pelas funções de inclusão, decidiu-se tratar um problema real.

A escolha do problema e do estudo de caso específico levou em consideração aspectos tais como: a) a abrangência da aplicação na engenharia; b) o interesse da comunidade científica na área; e c) a aplicabilidade de métodos intervalares. Observando estes critérios, decidiu-se pela otimização de processos controlados por PID (Proportional Integral Derivative). Nas próximas seções, uma formulação geral para o problema robusto multi-objetivo para sintonia de PID é apresentada; uma formulação específica para resolver o estudo de caso proposto é descrita juntamente com os detalhes de resolução pelos algoritmos [I]RMOA I, [I]RMOA II e [I]RMOEA. Antes das formulações, é feita uma revisão sobre os controladores PID e critérios de estabilidade.

### 5.1 Revisão sobre o Controlador PID

Os controladores PID, sejam eles analógicos ou digitais, são encontrados em diversas aplicações industriais como no controle de abastecimento e vazão de líquidos, no controle de temperatura e pressão de caldeiras e no controle de velocidade de motores, por exemplo. Astrom e Hugglund [5] justificam que pela simplicidade de funcionamento e com o desempenho muitas vezes considerado ótimo, o emprego de controladores PID na indústria chegou a atingir a cifra dos 95%, em se tratando de sistemas malha-fechada. A introdução de controladores do tipo PID tem como finalidade gerir processos com eficácia e precisão, tais que na ausência do controlador exigiriam atenção contínua de um ou mais operadores para realizar a tarefa de controle.

No controlador PID, o ajuste da variável de controle é feito de forma automática segundo às medidas obtidas na saída do processo  $z(t)$  comparadas a um dado valor de referência  $r(t)$ , ao longo do tempo  $t$ . A diferença entre os valores de  $r(t)$  e de  $z(t)$  é o *signal de erro*  $e(t)$

$$e(t) = r(t) - z(t), \quad (5.1)$$

que se deseja minimizar ou eliminar, estabilizando  $z(t)$  no ponto mais próximo de  $r(t)$ . A ação de controle age no processo de forma proporcional, integral e derivativa em relação ao sinal de erro conforme a expressão

$$m(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt}, \quad (5.2)$$

sendo  $m(t)$  a saída do controlador,  $K_p$  o ganho proporcional,  $K_i$  o ganho de integração e  $K_d$  o ganho derivativo. Um esquema simples de um controlador PID associado a um processo genérico ou uma *planta* está apresentado na Fig.5.1.

O vetor de parâmetros  $\mathbf{K} = \{K_p, K_i, K_d\}$  do PID é ajustado em função da planta a ser controlada. A variável a ser manipulada assume os valores definidos pelo controlador, guiado por  $e(t)$ . A tarefa de selecionar valores apropriados para os ganhos  $\mathbf{K}$  é chamada de *calibração ou sintonia de PID*. Existem diferentes métodos para calibração desses controladores, tais como: a) manualmente (tentativa-erro); b) pelo método de Ziegler-Nichols [129]; e c) pelo método Cohen-Coon [22]. Cada um desses métodos tem a sua aplicabilidade. Mais detalhes sobre sintonia de controladores podem ser encontrados em Visioli [122], Wang e Cai [125], e Chidambaram e Sree [19].

A ação dos ganhos  $\mathbf{K}$  no controlador é linear, sendo o papel de cada ganho no controle da planta bem definido. Em poucas palavras, o termo  $K_p$  acelera a resposta linear do sistema em direção ao valor de referência

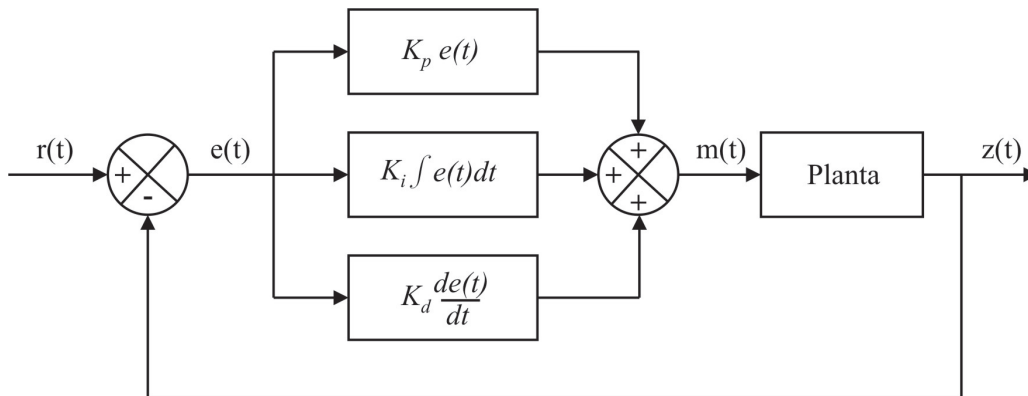


Fig. 5.1: Diagrama de blocos de um sistema malha-fechada com PID.

desejado. A integral relacionada ao ganho  $K_i$  computa a contribuição do erro acumulativo. Assim, o ganho  $K_i$  age sobre o erro conhecido ao longo do intervalo de integração. O termo derivativo contabiliza a variação do valor do erro no tempo. Portanto, o ganho simbolizado por  $K_d$  funciona como um compensador para a taxa de desvio do valor de referência. Importante salientar que existe uma relação de dependência entre os ganhos; a alteração do valor de um, interfere no valor a ser ajustado nos outros.

## 5.2 Descrição do Problema

A otimização de sistemas controladores PI, PD ou PID para uma planta genérica, significa encontrar melhor conjunto de parâmetros  $K_p$ ,  $K_i$  e  $K_d$  que minimizam algumas métricas/especificações de desempenho da resposta do sistema. As especificações no domínio do tempo são observáveis nos sistemas em que a resposta temporal é *estável*. A Fig.5.2 apresenta a identificação dessas métricas em um exemplo de resposta transitória de um sistema malha-fechada. O *sobre-sinal máximo*  $M_p$  é o valor mais alto da resposta transitória acima da referência. O *tempo de subida*  $t_s$  registra a velocidade com que o sistema responde à entrada, sendo portanto definido em intervalos de tempo proporcionais ao valor da entrada. Ogata [91] apresenta como valores típicos para  $t_s$ : a) de 10% a 90%, b) de 5% a 95% e c) de 0% a 100%. Para efeitos deste trabalho foi a faixa (c) para  $t_s$ . Similarmente, o *tempo de pico*  $t_p$  é medido como o tempo do início do processo até o valor de  $M_p$ . O *tempo de acomodação*  $t_a$  marca a estabilização da resposta do sistema numa faixa de valor estável (tolerância) próxima do valor de referência em que se deseja que o sistema atinja. A tolerância permitida é normalmente definida em torno de  $\pm 2\%$  a  $\pm 5\%$ . Segundo Ogata [91], exceto em certas aplicações em

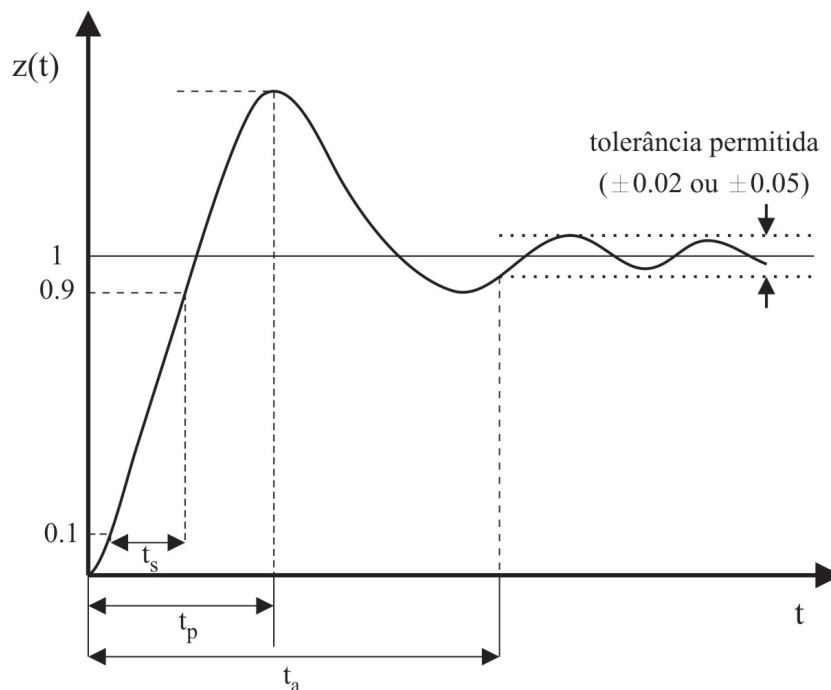


Fig. 5.2: Especificações para caracterizar a resposta transitória de um sistema de controle malha-fechada.

que não se pode aceitar oscilações, é desejável que a resposta transitória seja suficientemente rápida e amortecida. Portanto, considerando Ogata e as métricas da resposta transitória listadas acima, é desejável que o processo de otimização minimize  $M_p$ ,  $t_s$ ,  $t_p$  e  $t_a$ . Adicionalmente, foi escolhida uma métrica baseada no acúmulo do erro para contabilizar a medida da qualidade da resposta do sistema. Optou-se pela *métrica ISE* (Integral Square-Error) dada pela equação

$$ISE = \int_0^{\infty} e^2(t)dt. \quad (5.3)$$

As medidas de desempenho e qualidade descritas acima são utilizadas para formular o problema de otimização de sintonia de controladores PID. No entanto, antes de descrever o problema, são apresentadas condições em que a estabilidade é garantida.

### 5.2.1 Análise da Estabilidade do Sistema

A estabilidade é fundamental em sistemas de controle realimentados, devendo portanto ser tratada como um pré-requisito para o projeto de sistemas de

controle lineares. Por isso, decidiu-se por aplicar um método para verificar se o sistema otimizado é estável ou não. Os detalhes do método são apresentados a seguir.

*Cr terio de Routh-Hurwitz*

Para o caso de controle estudado, escolheu-se o m todo RH (Routh-Hurwitz). Em poucas palavras, usando apenas os coeficientes da equa o caracter stica, o m todo RH   conclusivo sobre a exist ncia ou n o de ra zes positivas da equa o polinomial caracter stica (denominador da fun o de transfer ncia de malha-fechada), sem que haja necessidade de se resolver a equa o. O m todo   apresentado considerando-se inicialmente a equa o caracter stica

$$a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0, \quad a_n \neq 0, \tag{5.4}$$

sendo os coeficientes grandezas reais. A partir de (5.4) constr i-se a Tab.5.1 apresentada a seguir.

Tab. 5.1: Coeficientes RH.

$s^n$	$a_n$	$a_{n-2}$	$a_{n-4}$	$\dots$
$s^{n-1}$	$a_{n-1}$	$a_{n-3}$	$a_{n-5}$	$\dots$
$s^{n-2}$	$b_1$	$b_2$	$b_3$	$\dots$
$s^{n-3}$	$c_1$	$c_2$	$c_3$	$\dots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Os coeficientes  $b_1, b_2, \dots$  s o calculados como

$$b_1 = \frac{a_{n-1}a_{n-2} - a_n a_{n-3}}{a_{n-1}}, \quad b_2 = \frac{a_{n-1}a_{n-4} - a_n a_{n-5}}{a_{n-1}}, \quad \dots \tag{5.5}$$

E os coeficientes  $c_1, c_2, \dots$  por

$$c_1 = \frac{b_1 a_{n-3} - b_2 a_{n-1}}{b_1}, \quad c_2 = \frac{b_1 a_{n-5} - b_3 a_{n-1}}{b_1}, \quad \dots \tag{5.6}$$

Pode-se concluir a partir do cr terio RH que o sistema   est vel se todos os termos da coluna iniciada por  $a_n$  na Tab.5.1 forem positivos ou se todos forem negativos. Esse cr terio foi definido para pontos reais. A seguir a interpreta o do cr terio RH para intervalos.

*Polinômios de Kharitonov*

O critério de RH pode ser estendido para tratar polinômios característicos cujos coeficientes sejam intervalos. No entanto, pode ser notado na Tab.5.1 que existem múltiplas ocorrências de coeficientes conduzindo o critério a um resultado muito conservativo. Uma alternativa para resolver o problema da estabilidade nesse caso é a aplicação da técnica de Kharitonov [70]. Para introduzir a descrição desta técnica, considere o seguinte polinômio característico intervalar

$$[a_n]s^n + [a_{n-1}]s^{n-1} + \dots + [a_1]s + [a_0], \quad 0 \notin [a_n], \quad (5.7)$$

e os seguintes polinômios de Kharitonov

$$p_1 : \bar{a}_n s^n + \underline{a}_{n_1} s^{n-1} + \underline{a}_{n-2} s^{n-2} + \bar{a}_{n-3} s^{n-3} + \bar{a}_{n-4} s^{n-4} + \dots \quad (5.8)$$

$$p_2 : \bar{a}_n s^n + \bar{a}_{n_1} s^{n-1} + \underline{a}_{n-2} s^{n-2} + \underline{a}_{n-3} s^{n-3} + \bar{a}_{n-4} s^{n-4} + \dots \quad (5.9)$$

$$p_3 : \underline{a}_n s^n + \bar{a}_{n_1} s^{n-1} + \bar{a}_{n-2} s^{n-2} + \underline{a}_{n-3} s^{n-3} + \underline{a}_{n-4} s^{n-4} + \dots \quad (5.10)$$

$$p_4 : \underline{a}_n s^n + \underline{a}_{n_1} s^{n-1} + \bar{a}_{n-2} s^{n-2} + \bar{a}_{n-3} s^{n-3} + \bar{a}_{n-4} s^{n-4} + \dots \quad (5.11)$$

O sistema é declarado estável se  $p_1, p_2, p_3$  e  $p_4$  forem estáveis, segundo o critério RH. Para sistemas de ordem  $n \leq 5$ , Anderson *et al.* [3] apresentam as seguintes simplificações nos polinômios de Kharitonov

$$n = 2 : \text{estável se } a_2, a_1, a_0 < 0 \text{ ou } a_2, a_1, a_0 > 0 \quad (5.12)$$

$$n = 3 : \text{estável se } p_1 \text{ for RH} \quad (5.13)$$

$$n = 4 : \text{estável se } p_1, p_2 \text{ forem RH} \quad (5.14)$$

$$n = 5 : \text{estável se } p_1, p_2, p_3 \text{ forem RH} \quad (5.15)$$

Muitos casos de sistemas de controle são formulados como sistemas de ordem até 5. Logo as simplificações de Anderson *et al.* representam ganho “expressivo” na computação de estabilidade das plantas.

*5.2.2 Formulação do Problema Multi-Objetivo Robusto para Sistemas de Controle com Planta Intervalar*

Em recente pesquisa sobre trabalhos que envolvam controle robusto de plantas intervalares, observou-se o crescente interesse da comunidade científica nesta linha de pesquisa, como por exemplo Dahleh *et al.* [25], Pujara e Roy [96]; e Henrion e Bachelier [55]. No contexto acima, termo “controle robusto” está relacionado à manutenção da estabilidade do sistema caso haja alterações nos valores dos parâmetros do sistema, dentro de uma região delimitada por intervalos. O termo “robusto” no contexto de otimização definido nesta

tese tem significado semelhante: encontrar o melhor conjunto de parâmetros tais que, dentro de uma região garantidamente estável, maximize o desempenho da resposta transitória. A seguir, apresenta-se a descrição sobre alguns trabalhos que envolvem otimização de controladores em plantas intervalares.

Hsu e Yu [59] descreveram um sistema de otimização mono-objetivo utilizando o ISE como métrica a minimizar. Um algoritmo genético foi empregado para realizar tal tarefa. Importante salientar que foi utilizada a filosofia do pior caso para otimização robusta, com o critério RH funcionando como função de restrição, sendo a incerteza considerada apenas nos parâmetros da planta; os parâmetros do controlador foram considerados com valores fixos. Lordelo *et al.* [77] usaram plantas intervalares para localizar os pólos do sistema. Partindo da função de transferência da planta e do polinômio intervalar característico do sistema malha-fechada, os *controladores intervalares* foram obtidos via resolução da equação de Diofantina intervalar. Esses controladores, no formato de intervalos, contêm todas as possibilidades de localização dos pólos. Na seqüência, os controladores foram empregados para caracterizar a região no plano que assegura o bom desempenho para o caso malha-fechada. Então, um método de otimização determinou os *controladores não frágeis*, objetivo principal do trabalho. Takahashi *et al.* [119] apresentaram um método para otimizar o sistema de controle PID de plantas com incerteza nos parâmetros e ainda sujeita a distúrbio. A otimização de desempenho do sistema foi formulada como problema multi-objetivo, cujos critérios foram definidos empregando-se as normas  $\mathcal{H}_2$  e  $\mathcal{H}_\infty$  da matriz de transferência dos distúrbios e da referência de entrada para o erro da saída controlada. Os autores garantiram a estabilidade pela inspeção de politopo construído a partir dos valores dos parâmetros com a respectiva tolerância.

Dentre os estudos apresentados acima, a aplicação proposta a seguir se assemelha: a) ao trabalho de Hsu e Yu [59], no que se refere ao tratamento do quesito estabilidade; b) ao artigo de Lordelo *et al.* [77], porque manipula os controladores como intervalos; e c) ao estudo de Takahashi *et al.* [119] quanto ao tipo de formulação multi-objetivo do problema. Assim, sendo o vetor  $\mathbf{K}$  os ganhos do sistema e o parâmetro  $\mathbf{p}$  a presença de incertezas, um problema robusto multi-objetivo para sintonia de controladores PID de uma

planta genérica de ordem  $n$  pode ser escrito como

$$\begin{aligned} \min_{\mathbf{K} \in [\mathbf{K}]} \max_{\mathbf{p} \in [\mathbf{P}]} f_1, f_2, f_3, f_4 \\ \text{sujeito a : } g_1 = \text{critério RH;} \\ g_2 = \max z(\mathbf{K}, \mathbf{p}, t) < \text{desvio aceitável de } u(t), t = t_0; \\ g_3 = \text{singularidades Análise Intervalar;} \\ g_4 = \text{outras singularidades.} \end{aligned} \quad (5.16)$$

$$\begin{aligned} \text{Sendo : } f_1 &= \max z(\mathbf{K}, \mathbf{p}, t), t \in [0, \infty]; \\ f_2 &= \arg \max_{t \in [0, \infty]} (f_1); \\ f_3 &= \min_{t \in [0, \infty]} \{t \mid z(\mathbf{K}, \mathbf{p}, t) = r(t)\}; \\ f_4 &= \text{ISE.} \end{aligned}$$

Normalmente o sobre-sinal máximo ( $f_1$ ) e o tempo de pico ( $f_2$ ) são especificações conflitantes, ou seja, o decréscimo do sobre-sinal acarreta no acréscimo do tempo de pico. O crescimento/decréscimo do tempo de subida ( $f_3$ ) está em conformidade com o crescimento/decréscimo de  $t_p$ . Porém, em algumas situações  $M_p$  e  $t_p$  podem não existir, pois nem sempre há sobre-sinal. Com isso as métricas  $M_p$  e  $t_p$  podem resultar em quesitos “ruins” a otimizar. Por outro lado, o índice ISE ( $f_4$ ) representa um quesito seguro a otimizar. O ISE contabiliza o menor erro acumulado no período transitório. Em (5.16), a restrição  $g_1$  está relacionada à verificação da estabilidade via Routh-Hurwitz, Kharitonov e simplificações de Anderson *et al.*, quando for o caso. A restrição  $g_2$  desqualifica as respostas “ruins” com erro em estado estacionário, até o tempo  $t = t_0$ . E,  $g_3$  certifica a aplicabilidade dos métodos intervalares. Por exemplo, considere o intervalo  $[a, b]$ ,  $a \in \mathbb{R}$  e  $b \in \mathbb{R}$ . Se  $a < 0$  e  $b > 0$ , os cálculos  $\sqrt{[a, b]}$  e  $1/[a, b]$  representam singularidades, pois não estão definidos para Análise Intervalar que emprega sistemas finitos, que é o caso desta Tese. Finalmente,  $g_4$  trata outras singularidades. Por exemplo, evita que a formulação gere resultados impossíveis tais como  $t_s < 0$ .

### 5.3 Estudo de Caso

A formulação proposta em (5.16) pode ser empregada em vários problemas de controle via PID. Por isso é necessário estabelecer o escopo para o estudo de caso abordado aqui. O caso considerado é simples. No entanto, a finalidade é

detalhar o processo de otimização robusta. O leitor interessado pode estender o procedimento para resolver sistemas cuja função de transferência tenha ordem  $n \leq 4$ . A limitação de ordem deve-se à necessidade da função de inclusão da resposta do sistema ser analítica. Como a resposta transitória depende do cálculo das raízes do polinômio característico, a solução analítica fechada fica restrita até equações de quarta ordem.

Para melhor descrição do estudo de caso, esta seção foi dividida nas seguintes partes: a) Caracterização da Planta; b) Resposta Temporal do Sistema; c) Formulação do Problema Multi-Objetivo Robusto; d) Funções Objetivo; d) Análise das Funções Objetivo e de Restrição; e e) Experimento e Resultados.

### 5.3.1 Caracterização da Planta

No estudo de caso, foi considerado um sistema de controle linear, invariante no tempo, sem atraso, sem distúrbio de entrada e com controle analógico proporcional-mais-derivativo para responder a uma entrada degrau unitário. O sistema está apresentado na Fig.5.3. A função de transferência resultante

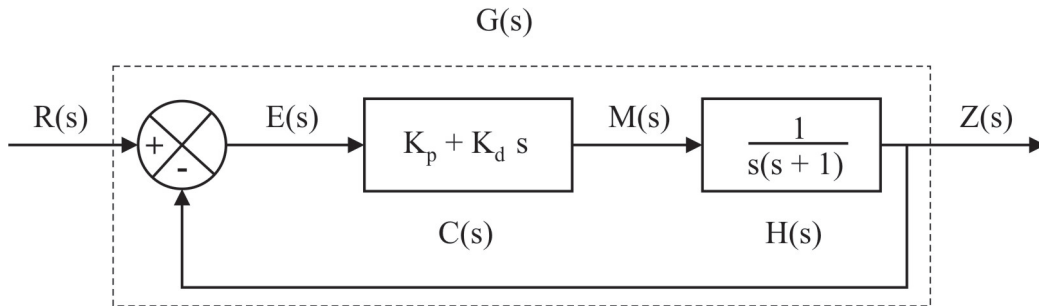


Fig. 5.3: Diagrama de blocos do estudo de caso.

$G(s)$  para o caso da Fig.5.3 é dada por

$$G(s) = \frac{H(s)C(s)}{1 + H(s)C(s)} = \frac{K_p + (K_d)s}{s^2 + (1 + K_d)s + K_p} \quad (5.17)$$

### 5.3.2 Resposta Temporal do Sistema

Aplicando o degrau unitário ( $r(t) = 0, t = 0$ ;  $r(t) = 1, t > 0$ ) e realizando a transformada inversa de Laplace obtém-se a seguinte resposta do sistema  $z(K_p, K_d, t)$

$$z(K_p, K_d, t) = 1 + e^{-\frac{(1+K_d)}{2}t} \left( -\cosh\left(\frac{\sqrt{\alpha}}{2}t\right) + \frac{(-1 + K_d)}{\sqrt{\alpha}} \sinh\left(\frac{\sqrt{\alpha}}{2}t\right) \right) \quad (5.18)$$

sendo

$$\alpha = (K_d + 1)^2 - 4K_p. \quad (5.19)$$

Considerando a incerteza  $p$  na precisão do controlador analógico, os ganhos  $K_p$  e  $K_d$  devem ser reescritos como  $K_p \pm p$  e  $K_d \pm p$ . A ação dos controladores  $K_p$  e  $K_d$  sobre a planta é linear, no entanto o efeito aditivo da incerteza  $p$  sobre  $K_p$  e  $K_d$  não é trivial. Assim, considerando  $p$ , as formulações em (5.18) e (5.19) podem ser reescritas como

$$\begin{aligned} z(K_p, K_d, p, t) &= 1 + e^{-\frac{(1+K_d+p)}{2}t} \left( -\cosh\left(\frac{\sqrt{\alpha'}}{2}t\right) \right) + \dots \\ &\dots + e^{-\frac{(1+K_d+p)}{2}t} \left( \frac{(-1 + K_d + p)}{\sqrt{\alpha'}} \sinh\left(\frac{\sqrt{\alpha'}}{2}t\right) \right) \end{aligned} \quad (5.20)$$

sendo

$$\alpha' = (K_d + p + 1)^2 - 4(K_p + p). \quad (5.21)$$

### 5.3.3 Formulação do Problema Multi-Objetivo Robusto

Considerado a resposta temporal do sistema descrita por (5.20) e (5.21), o problema robusto multi-objetivo para o estudo de caso é definido como

$$\begin{aligned} \min_{K_p, K_d \in [0, 15]} \quad \max_{p \in [-0.1, 0.1]} \quad & f_1, f_2, f_3, f_4 \\ \text{sujeito a :} \quad & g_1 = s^2 + (1 + K_d + p)s + (K_p + p) \text{ for RH;} \\ & g_2 = \max z(K_p, K_d, p, t) < 1.3, \quad t = 3s; \\ & g_3 = \alpha' > 1; \\ & g_4 = \frac{-\sqrt{\alpha'} + 1 - (K_d + p)}{\sqrt{\alpha'} + 1 - (K_d + p)} > 1, \quad 0 \notin \sqrt{\alpha'} + 1 - (K_d + p). \end{aligned} \quad (5.22)$$

$$\begin{aligned} \text{Sendo :} \quad & f_1 = \max z(K_p, K_d, p, t), \quad t \in [0, \infty]; \\ & f_2 = \arg \max_{t \in [0, \infty]} (f_1); \\ & f_3 = \min_{t \in [0, \infty]} \{t \mid z(K_p, K_d, p, t) = r(t)\}, \quad r(t) = 1; \\ & f_4 = \text{ISE.} \end{aligned}$$

### 5.3.4 Funções Objetivo

As funções objetivo  $f_1$  e  $f_2$  referem-se ao sobre-sinal máximo  $M_p$  e ao tempo de pico  $t_p$  da resposta do sistema  $z(K_p, K_d, p, t)$  descrita em (5.20). Para funções contínuas, a derivada tem valor nulo nos pontos de máximo e mínimo da função. Portanto,  $M_p$  e  $t_p$  ocorrem quando

$$\frac{dz(K_p, K_d, p, t)}{dt} = 0, \quad (5.23)$$

resultando em

$$t_p = \frac{1}{\sqrt{\alpha'}} \ln \left( \frac{1 - (\sqrt{\alpha'} + (K_d + p))^2}{1 - (\sqrt{\alpha'} - (K_d + p))^2} \right), \quad (5.24)$$

e

$$M_p = z(K_p, K_d, p, t_s) - 1. \quad (5.25)$$

As deduções das equações (5.24) e (5.25) foram realizadas no Apêndice A considerando um modelo sem o parâmetro de incerteza  $p$  e para valores fixos de  $K_p$  e  $K_d$ .

A função objetivo  $f_3$ , que trata do tempo de subida  $t_s$ , também é obtida da resposta do sistema  $z(K_p, K_d, p, t)$ . Considerando  $t_s$  como o período para o sistema atingir a referência ( $r(t) = 1$ ),  $t_s$  é formulado como

$$t_s = \frac{1}{\sqrt{\alpha'}} \ln \left( \frac{-\sqrt{\alpha'} + 1 - (K_d + p)}{\sqrt{\alpha'} + 1 - (K_d + p)} \right) \quad (5.26)$$

O desenvolvimento da equação para  $t_s$  foi realizado no Apêndice B, que sem perda de generalidade, foram fixados  $K_p$  e  $K_d$ , e o parâmetro  $p$  foi desconsiderado.

A função  $f_4$ , referente ao índice ISE, é calculada a partir da equação integral (5.3). A resolução da integral, feita no Apêndice C, produz como resultado

$$\begin{aligned} \text{ISE} = & -\frac{\left(1 + \frac{1-(K_d+p)}{\sqrt{\alpha'}}\right)^2}{\sqrt{\alpha'} - 1 - (K_d + p)} \cdots \\ & \cdots \frac{2\left(1 - \left(\frac{1-(K_d+p)}{\sqrt{\alpha'}}\right)^2\right)}{-1 - (K_d + p)} - \frac{\left(1 - \frac{1-(K_d+p)}{\sqrt{\alpha'}}\right)^2}{-\sqrt{\alpha'} - 1 - (K_d + p)}. \end{aligned} \quad (5.27)$$

Os algoritmos para otimização robusta propostos nesta tese utilizam-se de funções de inclusão intervalares para o cálculo do desempenho das soluções

candidatas. As funções de inclusão criadas para as funções objetivo do estudo de caso foram classificadas como funções de inclusão naturais, ou seja, foram obtidas pela substituição da variável real pela sua correspondente variável intervalar. Como as funções objetivo descritas pelas equações (5.24-5.27) contêm apenas operadores matemáticos elementares, pode-se afirmar que elas são convergentes e monotônicas. No entanto, a característica de ser mínima e mais estreita possível depende do número de ocorrências das variáveis internas e do tipo de operação em que essas ocorrências múltiplas estão envolvidas. Esse fato justifica as simplificações realizadas nos apêndices deste texto. Acredita-se que outras simplificações ainda são possíveis. Desta forma, decidiu-se por não afirmar que as funções de inclusão deduzidas são estreitas. Contudo, elas atendem a finalidade de suporte à otimização pelos algoritmos propostos.

### 5.3.5 Análise das Funções Objetivo e de Restrição

O tratamento das restrições da formulação (5.22) é similar àquele procedimento apresentado na Subseção 3.5.1. No tratamento de restrições deste estudo de caso, os testes de inclusão classificam todo o espaço de busca em “viável” e “de fronteira”, segundo dada precisão. Nenhum espaço foi classificado como “não viável” porque não foram adotadas funções de restrição que garantissem a instabilidade. A Fig.5.4 apresenta o espaço viável do estudo de caso que será objeto da tarefa de otimização.

Neste ponto, torna-se importante esclarecer como as funções de inclusão interpretam a resposta do sistema. Isso é feito através do exemplo que se segue. Considere a caixa viável  $[\mathbf{K}]_0$  formada por  $[K_p] + [p] = [9.3749, 10.3126]$  e  $[K_d] + [p] = [5.6249, 6.5626]$ . A Fig.5.5 apresenta o envelope criado a partir da função de inclusão natural  $[z]([K_p], [K_d], [p], t)$  de (5.20). A figura também mostra a resposta no ponto central desta caixa, ou seja, o ponto  $\mathbf{c}([\mathbf{K}]_0) = \{9.8438, 6.0938\}$ . Os resultados das funções de inclusão referentes às equações (5.24-5.27) para o exemplo acima descrito estão mostrados na Tab.5.2. Observe que as funções de inclusão para  $M_p$  e ISE geram

Tab. 5.2: Análise das funções objetivo de inclusão em  $[\mathbf{K}]_0$  e  $\mathbf{c}([\mathbf{K}]_0)$ .

	$M_p$	$t_p$ (s)	$t_s$ (s)	ISE
$[\mathbf{K}]_0$	[-644.5,2210.1]	[0.178, 3.495]	[0.104,2.447]	[-2.888,5.033]
$\mathbf{c}([\mathbf{K}]_0)$	1.0397	0.7737	0.4682	0.3106

resultados não estreitos e até mesmo irrealis, como por exemplo, o sobresinal não poderia ser negativo. No entanto, considerando a filosofia do pior

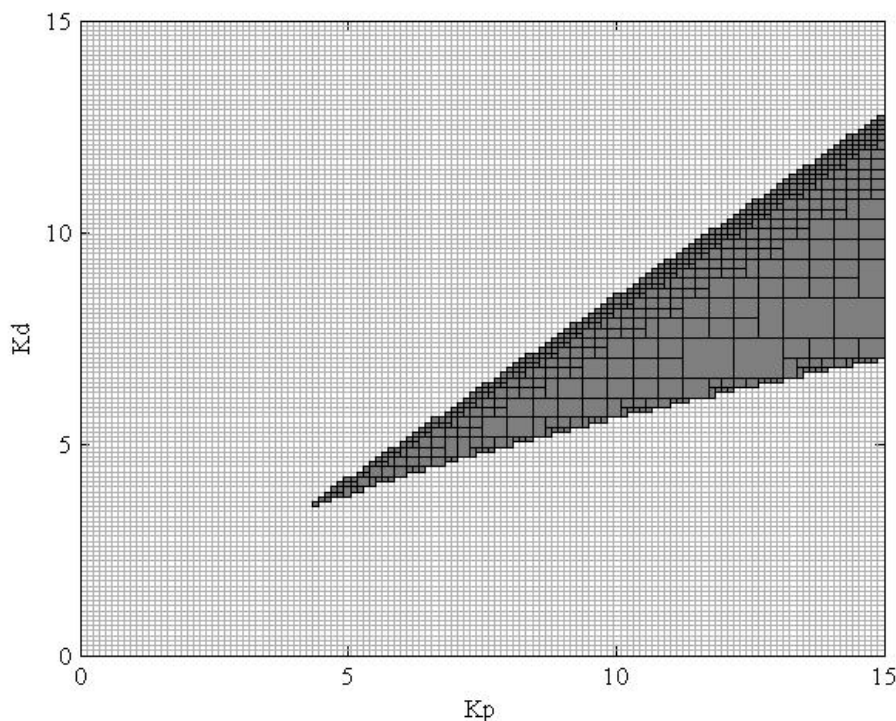


Fig. 5.4: Espaço de busca após o tratamento de restrições. A região em cinza escuro representa o espaço viável; em cinza claro, os intervalos que não puderam ser classificados como viáveis e nem como não viáveis. A figura foi obtida empregando-se o algoritmo SIVIA (2.3.12) para interpretar o resultado dos testes de inclusão das funções de restrição. Detalhes: parâmetro de precisão  $\varepsilon = 0.1$ ; 590 caixas viáveis.

caso, o que conta para efeitos de avaliação de robustez são os limites superiores. Para analisar a convergência e monotonicidade das funções de inclusão, considere as caixas  $[\mathbf{K}]_1 = [9.5749, 10.1126] \times [5.8249, 6.3626]$  e  $[\mathbf{K}]_2 = [9.7749, 9.9126] \times [6.0249, 6.1626]$  com as características  $[\mathbf{K}]_2 \subset [\mathbf{K}]_1 \subset [\mathbf{K}]_0$  e  $\mathbf{c}([\mathbf{K}]_2) = \mathbf{c}([\mathbf{K}]_1) = \mathbf{c}([\mathbf{K}]_0)$ . A Tab.5.3 mostra os resultados. Por outro lado, a Fig.5.6 mostra a convergência e monotonicidade da resposta do sistema.

A medida em que as bissecções vão sendo realizadas sobre as soluções candidatas, as funções de inclusão vão aumentando sua precisão auxiliando os algoritmos a aproximarem-se do ponto que eles realmente devem representar. No entanto, as bissecções geram um custo computacional que se eleva exponencialmente com as dimensões do problema e com a precisão desejada.

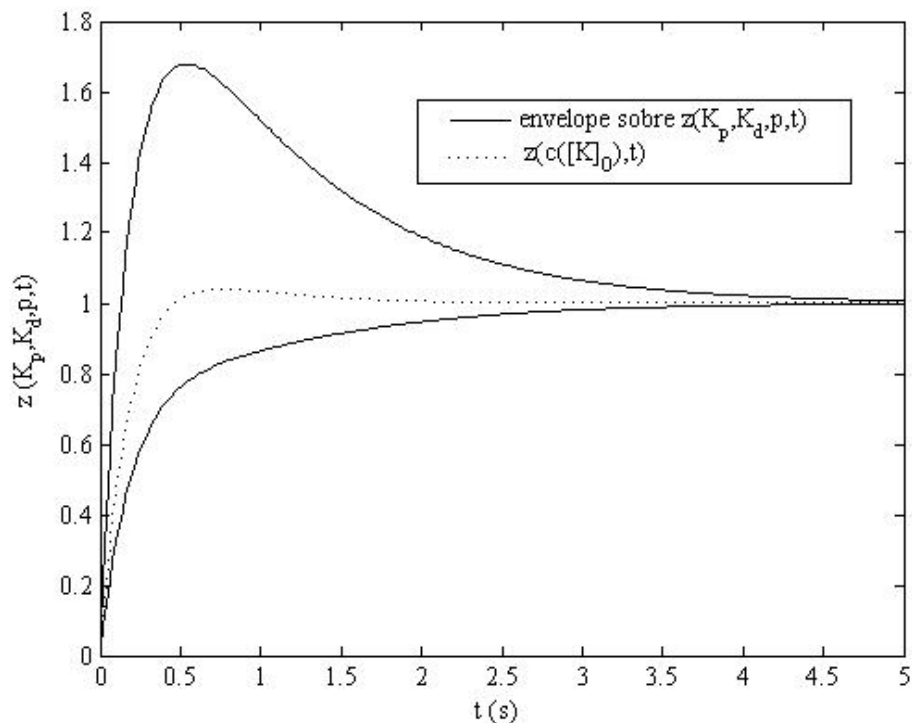


Fig. 5.5: Envelope sobre  $z(K_p, K_d, p, t)$ . A função de inclusão  $[z]([K_p], [K_d], [p], t)$  engloba todas as possíveis respostas contidas nos intervalos especificados.

Tab. 5.3: Análise das funções objetivo de inclusão em  $[K]_1$  e  $c([K]_1)$ .

	$M_p$	$t_p$ (s)	$t_s$ (s)	ISE
$[K]_0$	[-644.5,2210.1]	[0.178, 3.495]	[0.104,2.447]	[-2.888,5.033]
$[K]_1$	[-1.627,6.718]	[0.394, 1.536]	[0.233,0.975]	[-0.643,1.432]
$[K]_2$	[0.920,1.231]	[0.658, 0.911]	[0.395,0.556]	[0.108,0.520]
$c([K]_0)$	1.0397	0.7737	0.4682	0.3106

### 5.3.6 Experimento e Resultados

O experimento consistiu em utilizar os algoritmos [I]RMOA I, [I]RMOA II e [I]RMOEA, validados no Capítulo 4, para resolver o problema robusto multi-objetivo descrito em (5.22). Como trata-se de um problema com quatro objetivos a visualização gráfica ficou restrita para três objetivos simultâneos. Os resultados da otimização de (5.22) foram divididos em quatro grupos: a) envelopes sobre a fronteira robusta; b) conjuntos solução; c) fronteiras

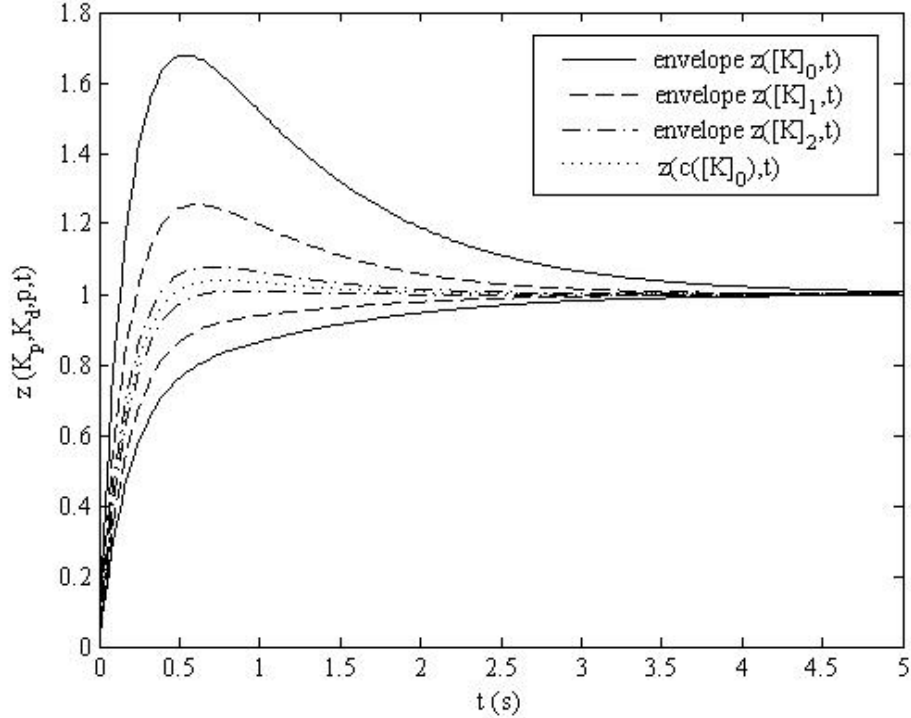


Fig. 5.6: Conjunto de envelopes sobre  $z(K_p, K_d, p, t)$ . Note as propriedades de convergência e monotonicidade da função de inclusão.

robustas; e d) análise de algumas soluções na resposta do sistema de controle. Em (a), os resultados no espaço dos objetivos dos três algoritmos são confrontados, mas o foco está nos envelopes. Em (b), os conjuntos solução encontrados via os algoritmos [I]RMOA II e [I]RMOEA são apresentados. Em (c), as fronteiras robustas do [I]RMOA II e do [I]RMOEA são traçadas e comparadas. E em (d), algumas soluções resultantes de [I]RMOA II e de [I]RMOEA são utilizadas como respostas para a sintonia do sistema de controle.

Para as execuções, conforme apresentado na formulação (5.22), foram empregados  $K_p = [0, 15]$ ,  $K_d = [0, 15]$  e  $P = [-0.1, 0.1]$ . Além destes parâmetros, foram configurados os seguintes parâmetros específicos: a)  $\varepsilon_{x\#} = 0.25$  e  $\varepsilon_{[x]} = 0.05$  para o [I]RMOA I; b)  $\varepsilon_{[x]} = 0.05$  e  $\varepsilon_{[p]} = 0.025$  para o [I]RMOA II; e c)  $p_c = 1$ ,  $p_m = 0.0005$ ,  $n_{pop} = 100$ ,  $n_{gen} = 60$ ,  $n_{bit} = 15$ ,  $\varepsilon_{[p]} = 0.025$  e  $r_{bis}^{max} = 8$  para o [I]RMOEA. No experimento que investiga os envelopes, também foram utilizados  $\varepsilon_{[p]} = 0.005$  e  $\varepsilon_{[p]} = 0.001$  para o [I]RMOA II e o [I]RMOEA.

Envelopes sobre a Fronteira Robusta

A visualização de envelopes 3D não é clara, por isso optou-se por apenas marcar as regiões de  $\mathbf{K}^-$  e  $\mathbf{K}^+$  pelos pontos que definem cada região e contrastar com as fronteiras robustas dos outros algoritmos. Os resultados estão apresentados nos gráficos que se seguem.

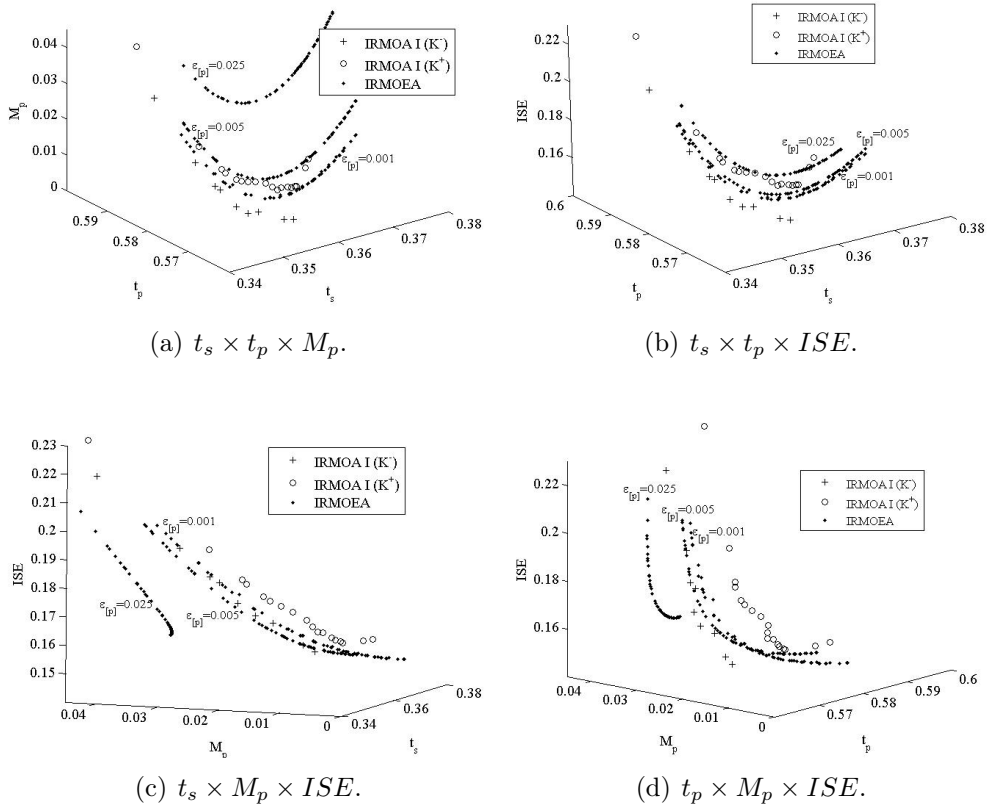


Fig. 5.7: Estudo de Caso PD - [I]RMOA I  $\times$  [I]RMOEA. O envelope definido por  $\mathbf{K}^-$  e  $\mathbf{K}^+$  reprovou as imagens robustas geradas pelo [I]RMOEA quando foram utilizados  $\varepsilon_{[p]} = 0.005$  e  $\varepsilon_{[p]} = 0.025$ .

A Fig.5.7 mostra a dependência da fronteira robusta com relação à precisão  $\varepsilon_{[p]}$  do parâmetro de incerteza  $\mathbf{p}$ . Portanto, uma ou mais funções de inclusão dos objetivos não são mínimas (veja Seção 2.3.8 para mais informações sobre funções de inclusão mínimas). Dentre as quatro sub-figuras, o envelope exibido no gráfico (b) envolveu tanto o resultado obtido com  $\varepsilon_{[p]} = 0.001$ , quanto  $\varepsilon_{[p]} = 0.005$ . Isto sugere que  $M_p$ , única função não plotada em (b), é função objetivo menos estreita. A confirmação desta in-

formação pode ser verificada resolvendo-se o problema formulado em (5.22) com menor número de objetivos.

Similarmente à figura anterior, a Fig.5.8 compara os envelopes com o conjunto solução gerado no [I]RMOA II.

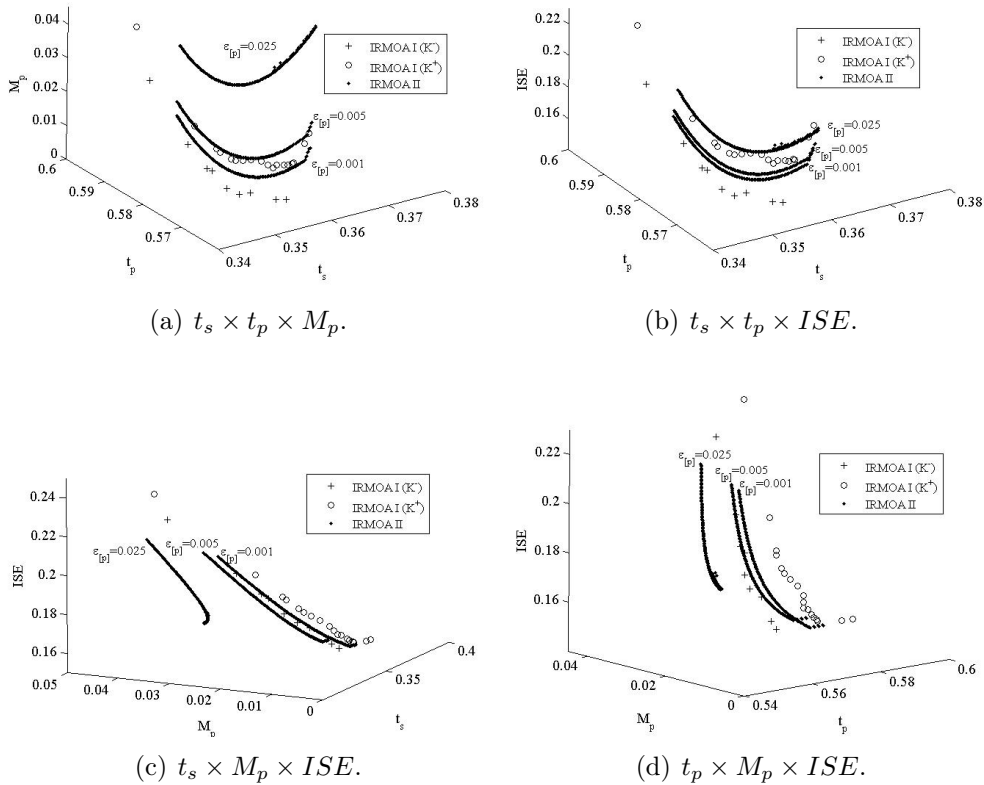


Fig. 5.8: Estudo de Caso PD - [I]RMOA I  $\times$  [I]RMOA II. O envelope definido por  $\mathbf{K}^-$  e  $\mathbf{K}^+$  mostrou conformidade do conjunto Pareto robusto gerado pelo [I]RMOA II quando foi empregado  $\varepsilon_{[p]} = 0.001$ . As fronteiras robustas para os demais valores de  $\varepsilon_{[p]}$  foram reprovadas.

A Fig.5.8 apresentou resultados semelhantes àqueles exibidos na Fig.5.7. Essa convergência de resultados, para esse estudo de caso, prova que: a) as funções objetivo, e conseqüentemente a fronteira de Pareto robusta, são dependentes de  $\varepsilon_{[p]}$ ; e b) o [I]RMOA I cumpriu o papel de ferramenta de verificação, reprovando as execuções com  $\varepsilon_{[p]}$  inadequado.

*Espaço de Busca*

Os conjuntos solução do problema robusto multi-objetivo em (5.22) determinados pelos métodos [I]RMOA II e [I]RMOEA encontram-se na Fig.5.9 e na Fig.5.10 respectivamente. Ao todo, foram geradas 181 ( $\varepsilon_{[p]} = 0.025$ ), 168 ( $\varepsilon_{[p]} = 0.005$  e  $\varepsilon_{[p]} = 0.001$ ) soluções pelo [I]RMOA II. E, 100 soluções para o [I]RMOEA, sendo esse número igual ao tamanho da população utilizada. Os resultados na Fig.5.9 e na Fig.5.10 sugerem que o ganho  $K_p$  ótimo tem valor 15 e que  $K_d$  ótimo se situa dentro do intervalo  $[9.5, 13.5]$ . Registra-se que o [I]RMOA II encontrou o mesmo conjunto solução para  $\varepsilon_{[p]} = 0.005$  e  $\varepsilon_{[p]} = 0.001$ .

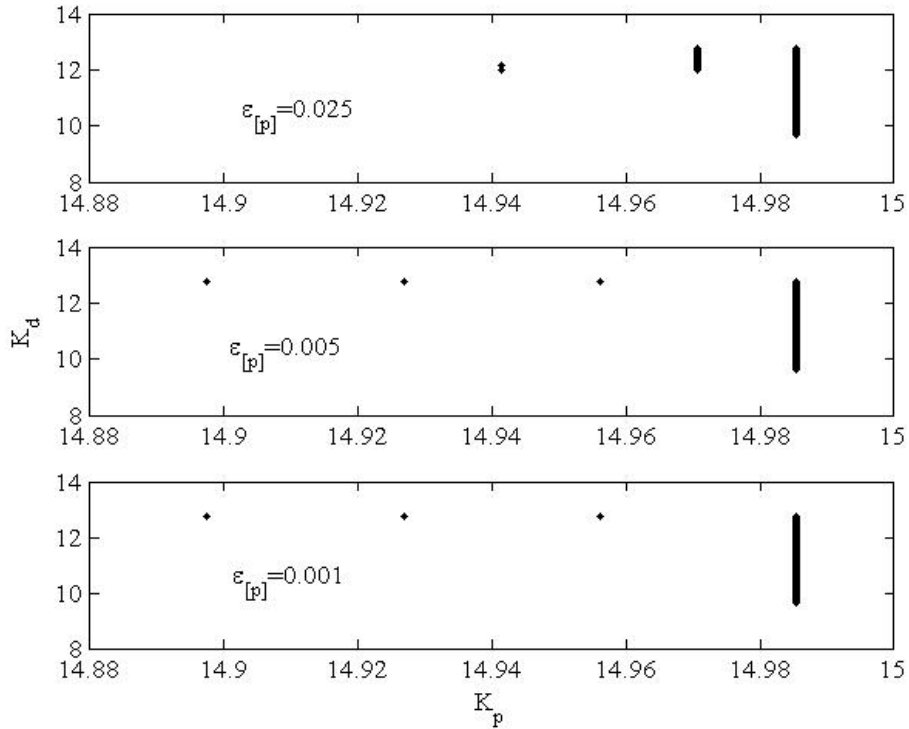


Fig. 5.9: Conjuntos solução do [I]RMOA II para o Estudo de Caso PD. A regularidade e proximidade da distribuição no conjunto solução do [I]RMOA II impede a visualização da distribuição das soluções. No entanto, é possível verificar que a precisão  $\varepsilon_{[x]} = 0.05$  foi atendida para a maioria das soluções.

Nota-se que apenas alguns elementos do conjunto solução [I]RMOA II que estão isolados à esquerda, não satisfizeram a precisão  $\varepsilon_{[x]} = 0.05$  definida como parâmetro no método. Não se trata de erro no método ou perda de

garantia. Por causa da estratégia pessimista adotada, uma região no espaço de busca só pode ser eliminada quando houver certeza que ela não contém solução robusta. Assim, os intervalos que contêm as citadas soluções isoladas ainda são candidatos a proverem soluções robustas. Em uma suposta próxima rodada de bissecções, o pessimismo é diminuído e as soluções isoladas tendem a desaparecer, deixando a fronteira gerada pelo [I]RMOA II com a distribuição ainda mais homogênea.

Pela Fig.5.10, nota-se que a dispersão do conjunto solução aumenta à medida que  $\varepsilon_{[p]}$  diminui. Dentre as possíveis causas, cita-se que o aumento da precisão na computação das funções objetivos de inclusão permitiu que soluções próximas à região ótima apresentarem desempenho equivalente.

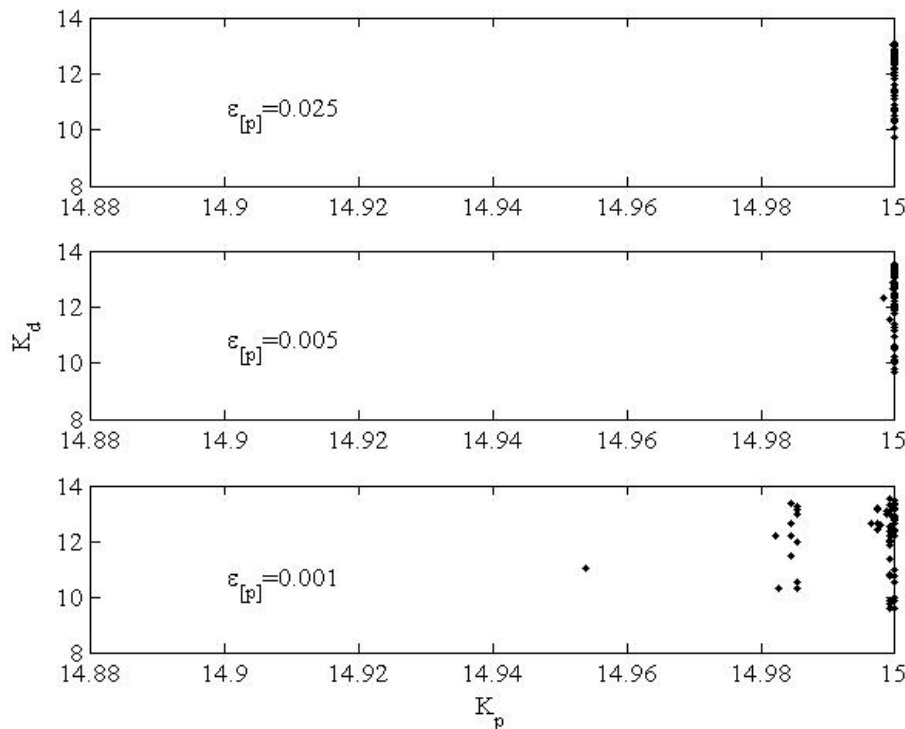


Fig. 5.10: Conjuntos solução do [I]RMOEA para o Estudo de Caso PD. Note que o número de soluções com  $K_p = 15$  diminui com o decréscimo de  $\varepsilon_{[p]}$ .

Observa-se que os conjuntos solução apresentados na Fig.5.9 e na Fig.5.10 convergiram para a região ótima independente do valor de  $\varepsilon_{[p]}$ . Por outro lado, a imagem robusta desses conjuntos solução ocupou regiões diferentes no espaço dos objetivos (veja a Fig.5.7 e a Fig.5.8). Essa informação é útil, pois permite reduzir o custo da computação das funções de inclusão com emprego

de  $\varepsilon_{[p]}$  maiores.

Comparações entre Fronteiras Robustas

Para um dado valor para  $\varepsilon_{[p]}$ , os métodos [I]RMOA II e [I]RMOEA produziram fronteiras de Pareto robustas em regiões “próximas” no espaço dos objetivos, como pode ser observado na Fig.5.11. Como os métodos funcionam utilizando-se de processos de busca diferentes, essa coerência de resultados comprova a equivalência entre os métodos para resolver o problema robusto e que os conjuntos solução gerados situam-se realmente na região ótima.

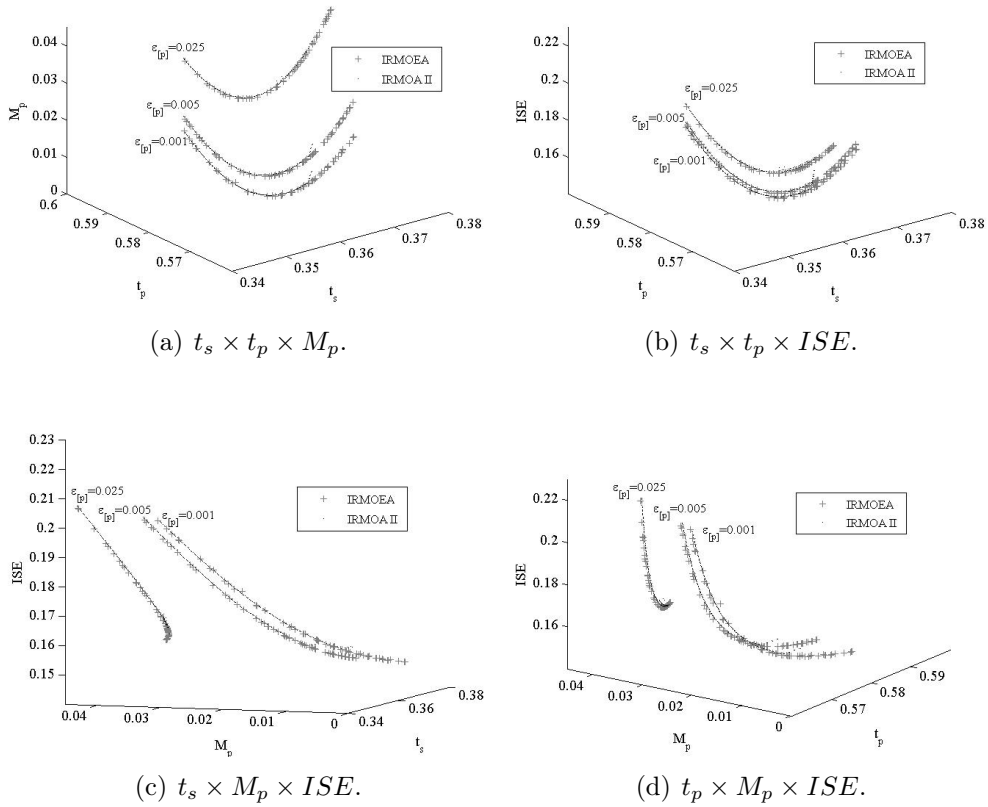


Fig. 5.11: Comparações entre as fronteiras robustas do [I]RMOA II e do [I]RMOEA. Observa-se que os métodos produzem fronteiras equivalentes para os mesmos valores de  $\varepsilon_{[p]}$ .

A Fig.5.11 apresenta distribuição do conjunto solução mais homogênea em [I]RMOA II que em [I]RMOEA. Por outro lado, a aproximação da fronteira robusta é melhor no [I]RMOEA, como pode ser visto na Fig.5.9 e na Fig.5.10.

Sintonia Ótima

Conforme a filosofia *a posteriori* (Subseção 2.1.3) empregada, durante o processo de otimização nenhuma preferência foi atribuída às funções objetivo. Portanto, o problema de otimização multi-objetivo proposto em (5.22) não pode ser considerado como concluído com a apresentação dos conjuntos solução e de suas respectivas fronteiras de Pareto robusta nesta seção. Para finalizar é necessário realizar a tomada de decisão, ou seja, definir a *solução robusta de projeto*  $\mathbf{K}^{**} = \{K_p^{**}, K_d^{**}\}$  que sintoniza otimamente o controlador. Dois  $\mathbf{K}^{**}$  foram escolhidos: um do conjunto solução do [I]RMOA II e outro do [I]RMOEA. O critério adotado para a tomada de decisão foi a mínima soma das funções objetivos normalizados. Assim, denotando o conjunto de soluções robustas por  $\mathbf{K}^*$  e por  $f_i^{\max}(\mathbf{K}^*, \mathbf{p})$  o máximo valor de desempenho da função objetivo  $f_i$  segundo  $\mathbf{K}^*$ ,  $\mathbf{K}^{**}$  é formulado como

$$\mathbf{K}^{**} = \arg \min_{\mathbf{k}^* \in \mathbf{K}^*} \sum_{i=1}^{n_f} \frac{f_i^{\max}(\mathbf{K}^*, \mathbf{p}) - f_i(\mathbf{k}^*, \mathbf{p})}{f_i^{\max}(\mathbf{K}^*, \mathbf{p})}, \quad \mathbf{p} \in \mathbf{P}, \quad (5.28)$$

sendo  $f_i^{\max}(\mathbf{K}^*, \mathbf{p}) > 0$  garantido em (5.24)-(5.27). Assim, considerando a precisão  $\varepsilon_{[p]} = 0.001$ , a solução robusta selecionada pelo [I]RMOA II foi  $K_p^{**} = 14.9854$  e  $K_d^{**} = 9.6240$ , e para o [I]RMOEA foi  $K_p^{**} = 15.0000$  e  $K_d^{**} = 9.6088$ . As respostas do sistema estão mostradas nas figuras que se seguem. De acordo com as métricas de desempenho do sistema de con-

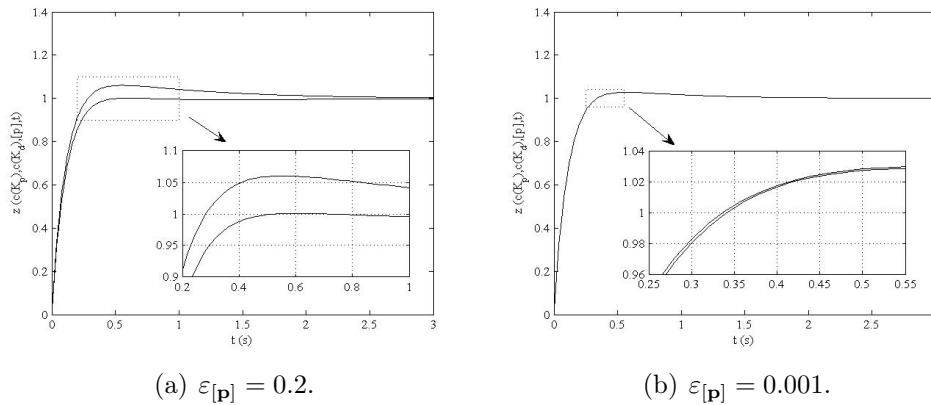


Fig. 5.12: Resposta do sistema para sintonia ótima segundo o [I]RMOA II. Em cada gráfico, a resposta robusta é curva superior. À esquerda, a incerteza é computada com um único cálculo,  $[p] = [-0.1, 0.1]$ ; À direita, o parâmetro de incertezas é subdividido em subpavimentos com precisão  $\varepsilon_{[p]} = 0.001$ . O envelope com  $\varepsilon_{[p]} = 0.001$  é mais preciso.

trole formuladas em (5.24)-(5.27), a solução do [I]RMOA II ( $K_p^{**} = 14.9854$ ,  $K_d^{**} = 9.6240$ ) resultou em  $t_s = 0.3411$ ,  $t_p = 0.5730$ ,  $M_p = 0.0305$  e  $ISE = 0.2031$ , e o [I]RMOEA ( $K_p^{**} = 15.0000$ ,  $K_d^{**} = 9.6088$ ) em  $t_s = 0.3409$ ,  $t_p = 0.5728$ ,  $M_p = 0.0307$  e  $ISE = 0.2034$ . Considerando a precisão de cada método, pode-se afirmar que os algoritmos convergiram para a mesma solução robusta, e por isso não foi necessário apresentar a resposta temporal [I]RMOEA. Observe na Fig.5.12 que a sintonia robusta ótima realmente produziu uma resposta eficiente ao sistema: estável, rápida, com sobre-sinal quase nulo e com resposta em regime permanente dentro da tolerância e, o mais importante, considerando as incertezas da formulação.

#### 5.4 *Considerações Finais*

A resolução de um problema de engenharia pelos algoritmos propostos representa um papel importante nesta tese: verificar como os problemas reais são tratados e resolvidos pelos algoritmos [I]RMOA I, [I]RMOA II e [I]RMOEA. Para atingir essa finalidade, decidiu-se por detalhar o processo de otimização, conforme listado a seguir. O procedimento de otimização foi inicializado com a contextualização do problema, destacando-se algumas abordagens interessantes encontradas em trabalhos científicos da área. Em seguida, foram apresentadas as métricas a serem otimizadas, as restrições a serem respeitadas e como pode ser inserido o parâmetro de incerteza. Como resultado, foi desenvolvida uma formulação robusta multi-objetivo que atendesse a diversos problemas de sintonia de controladores PID. Um estudo de caso, foi escolhido e as equações específicas para as funções objetivo de restrição foram desenvolvidas. O comportamento das funções de inclusão tanto aquelas relativas aos objetivos quanto as demais endereçadas às restrições foi discutido. Finalmente os algoritmos intervalares foram aplicados e os resultados obtidos foram confrontados, comprovando o desempenho de cada um dos métodos. A seguir as observações sobre os fatos mais importantes ocorridos durante o processo.

A resposta temporal de um sistema de controle genérico, conforme ilustrado na Fig.5.1, pode ser obtida facilmente utilizando-se de funções computacionais já implementadas em ferramentas computacionais comerciais, tais como a função “step” do Matlab. Como estas funções utilizam-se de métodos numéricos para tratar a resposta temporal, elas não podem ser utilizadas pelos algoritmos intervalares propostos, pois usualmente os métodos numéricos acarretam na perda de garantia de convergência e monotonicidade das funções de inclusão. Como exemplo, reveja novamente as características de convergência e monotonicidade na Fig.5.6 da resposta temporal do

sistema. No entanto, a resposta temporal no formato de função de inclusão ainda não atendeu às necessidades de interpretação do problema de otimização. Por exemplo, considere a resposta temporal  $[z](\mathbf{K}_0, t)$  apresentada na Fig.5.5. O sobre-sinal máximo relativo a todos os vetores contidos em  $[\mathbf{K}]_0$  poderia ser interpretado como a máxima amplitude mostrada na figura. Mesmo que essa interpretação não correspondesse à realidade, ela ainda poderia ser aceita como uma definição do projetista. O problema de obtenção das métricas via análise gráfica se tornaria mais difícil para determinar o “menor” sobre-sinal máximo de todos os vetores contidos em  $[\mathbf{K}]_0$ . Se o sobre-sinal máximo é uma métrica intervalar logo, ele deve ser definido por limites inferior e superior. Assim, os métodos intervalares também têm dificuldades em utilizar funções objetivos baseadas em valores medidos do sistema de otimização. Os métodos intervalares são mais facilmente aplicáveis quando a solução analítica das funções objetivo e de restrição forem disponíveis. Por isso, a resolução do problema de sintonia de controladores PID pelo procedimento descrito se limita a casos cuja equação polinomial característica seja de até quarto grau. A limitação deve-se ao fato da necessidade de se encontrar as raízes da equação característica pelo método analítico, que por sua vez é válido até a quarta ordem.

As restrições foram projetadas conforme as particularidades dos métodos intervalares e das métricas, e também para atender os requisitos de desempenho associados à planta. Uma grande vantagem dos métodos intervalares em relação a diversos outros métodos é o tratamento de restrições. As regiões não viáveis são excluídas permanentemente após a reprovação por alguma função de inclusão de restrição. Isso é sempre verdade quando as funções de restrição são conclusivas, o que não ocorreu na aplicação escolhida. O critério RH é um exemplo. Os intervalos eram testados quando à estabilidade. Os intervalos aprovados pelo critério eram selecionados; os demais eram continuamente bisseccionados, testados novamente pelo critério e reclassificados, até uma dada precisão. Uma função de inclusão mais eficiente para esse problema mediria a “instabilidade”. Assim, as soluções consideradas instáveis poderiam ser eliminadas mais eficientemente.

Os algoritmos [I]RMOA I, [I]RMOA II e [I]RMOEA foram considerados aprovados para resolver a aplicação proposta, pois mostraram-se eficazes tanto para formar um envelope ([I]RMOA I) sobre a fronteira robusta quanto para elencar vários pontos dessa fronteira ([I]RMOA II e [I]RMOEA).

## 6. CONCLUSÕES E TRABALHOS FUTUROS

A motivação principal desse trabalho foi resolver problemas de otimização em situações onde pode-se constatar a presença de incertezas. Decidiu-se por desenvolver métodos de busca que acoplassem as disciplinas Análise Intervalar, Otimização Robusta Multi-Objetivo e Algoritmos Evolucionários. Em separado, cada um dos tópicos acima representa uma grande área de pesquisa; a união deles certamente resulta em um amplo tema multidisciplinar. Por isso, fugiu completamente do escopo desta tese (Seção 1.3) a tentativa de esgotar qualquer um destes assuntos. Pelo contrário, a contribuição principal desta tese foi pontual: discutir otimização robusta multi-objetivo e introduzir ferramentas para resolvê-la cuja estrutura se baseia em métodos intervalares e evolucionários. Para finalizar esse estudo, nas seções que se seguem foram apresentadas as principais conclusões percebidas no decorrer da tese e alguns tópicos a serem cobertos em trabalhos futuros.

### 6.1 *Conclusões*

A escolha de estudar e propor algoritmos para tratar a filosofia de otimização robusta que considera o pior caso está ligada à decisão de se trabalhar com Análise Intervalar; a computação completa das incertezas é feita naturalmente nos métodos intervalares. Tratando as incertezas do sistema de otimização como um parâmetro separado, a computação pode ficar mais intuitiva. Se o parâmetro de incerteza aparece apenas uma vez na expressão da função objetivo robusta, com apenas uma avaliação da função objetivo pode-se computar toda incerteza com o mínimo de pessimismo (veja Subseção 3.5.1). Quando há múltiplas ocorrências do parâmetro de incerteza, então é necessária a criação de subpavimentos no espaço das incertezas para avaliar mais precisamente o desempenho robusto de dada solução. Vale a pena registrar que existem vantagens e desvantagens de se considerar o pior caso. Por exemplo, em otimização de dimensionamento de dispositivos a imagem de uma solução robusta pode, freqüentemente, ser um ponto fora do espaço dos objetivos. Em outras palavras, uma dada imagem robusta pode estar relacionada a pontos não pertencentes ao espaço de busca definido na formulação do problema. A situação piora com o aumento da ordem do es-

paço de busca. Por outro lado, situações onde não se pode correr risco, o pior caso deve ser a filosofia adotada para a otimização robusta.

O problema de otimização robusta é mais complexo que a otimização tradicional, pois considerando que as funções objetivo foram definidas como  $\mathbf{f}(\mathbf{x}, \mathbf{p}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_f}$ ,  $\mathbf{x} \in \mathbb{R}^{n_x}$  e  $\mathbf{p} \in \mathbb{R}^{n_p}$ , nota-se que a ordem do problema cresce com o fator  $n_x \times n_p$ . Se o sistema de otimização robusta envolver apenas funções objetivos convexas, a resolução é simples, rápida e precisa através da aplicação de métodos dirigidos por derivadas [13] (pg.13). No entanto, na fase de planejamento da tese, foi decidido desenvolver métodos para otimização robusta que fossem independentes de particularidades das funções objetivo e de restrições, como por exemplo não convexidade e descontinuidade da fronteira robusta.

Os métodos propostos para otimização robusta multi-objetivo [I]RMOA I, [I]RMOA II e [I]RMOEA apresentam, nesta ordem, a evolução temporal da criação dos algoritmos nessa tese. Cada um desses algoritmos teve sua contribuição. Primeiramente, a compreensão de diversos aspectos da fronteira de Pareto robusta partiu dos resultados do [I]RMOA I, através da construção de envelopes sobre a fronteira robusta. Uma vez que o [I]RMOA I é calibrado para dado valor de  $\mathbf{P}$ , o método está preparado para gerar outro envelope em um cenário robusto diferente de  $\mathbf{P}$ . Por outro lado, o [I]RMOA II busca as soluções robustas a partir de uma caixa contendo toda a região do espaço das variáveis. A região inviável é eliminada pelas funções de restrição, escritas também sob a forma intervalar. Utilizando uma filosofia pessimista, as regiões viáveis somente são descartadas quando puder ser provado que aquela região não contém soluções robustas. Assim, o ponto forte do [I]RMOA II é a garantia de armazenar toda a fronteira Pareto robusta, segundo dada precisão, dentro dos intervalos não excluídos. Tanto [I]RMOA I quanto [I]RMOA II sofrem com o aumento das dimensões no espaço de busca e de incerteza, porque são nesses espaços que são feitas as bissecções. Diferentemente, no [I]RMOEA a busca é probabilística e a influência das dimensões não causa aumento considerável no custo computacional. No entanto, o [I]RMOEA não garante encontrar soluções robustas. A contribuição dos métodos intervalares se restringe à computação da ação das incertezas sobre a formulação e durante a aplicação da técnica de nichos NB[I]. Dentre outros aspectos, uma característica verificada nos algoritmos [I]RMOA I e [I]RMOA II, e comum nos métodos puramente intervalares, é a convergência do mecanismo de busca. Isso pode ser facilmente verificado uma vez que todos os processos são formados por passos finitos. No [I]RMOA I, o processo de fatiar o espaço de busca ou dos objetivos para gerar as amostras que formarão  $\mathbf{K}^-$  e  $\mathbf{K}^+$  é finito por definição. O procedimento de busca de minimizadores para as amostras geradas (funções FPS e CSC) tem con-

vergência comprovada em [64]. No [I]RMOA II, o término do processo de busca também é certo e previsível; o mecanismo tem início com uma única caixa, e ao longo das iterações, as bissecções vão ocorrendo até que todas as caixas atinjam um valor mínimo de largura. Por outro lado, no [I]RMOEA a convergência é imposta fixando-se o número de iterações. Alternativamente, critérios de parada podem ser definidos como alguma métrica que mede a diversidade, por exemplo. De qualquer forma, o [I]RMOEA finaliza a busca por soluções sem a certeza que esgotou todas as possibilidades de busca. Finalmente, os métodos [I]RMOA I e [I]RMOA II têm apenas dois parâmetros, ao passo que [I]RMOEA tem vários. Essas características dos métodos desenvolvidos nesta tese concordam com a natureza puramente determinística do [I]RMOA I e do [I]RMOA II, e estocástica-determinística do [I]RMOEA. A Tab.6.1 resume a comparação dos algoritmos propostos. Importante observar que quesitos como “custo” tem avaliação subjetiva, levando em conta a análise empírica realizada na fase de testes.

Tab. 6.1: Comparação dos métodos [I]RMOA I, [I]RMOA II e [I]RMOEA. O símbolo † denota uma relação de dependência do método em relação aos itens listados.

Propriedade	[I]RMOA I	[I]RMOA II	[I]RMOEA
Classificação	determinístico	determinístico	híbrido
Busca	global	global	global
Finalidade	envelocar $\mathbf{Y}^*$	envelocar $\mathbf{X}^*$	encontrar $\mathbf{X}^*$
Qualidade $\mathbf{Y}^*$	não se aplica	alta	média/alta
Convergência	sim	sim	critério de parada†
Garantia	$\{\mathbf{f}, \varepsilon_{\mathbf{x}\#}, \varepsilon_{[\mathbf{x}]}\}^\dagger$	sim	não
Custo	alto	médio/alto	baixo
Dimensionalidade	sensível	sensível	pouco sensível
Parâmetros	poucos	poucos	muitos

O sucesso da uniformidade da fronteira robusta para o [I]RMOEA deve-se à técnica de nicho NB[I]. Pelos experimentos realizados, não há dúvidas quanto a sua eficiência e ao seu baixo custo computacional.

A métrica MB[I] mostrou-se eficaz na contabilização da uniformidade da fronteira robusta. Compare novamente os resultados na Subseção 3.7.2. Entretanto, ela foi desenvolvida para espaço dos objetivos com apenas duas dimensões.

Os métodos intervalares para otimização precisam construir funções de inclusão convergentes, monotônicas, mínimas e estreitas. Uma vez que esses quesitos sejam atendidos, os métodos intervalares podem excluir seguramente regiões onde a solução do sistema de otimização não está. Na verdade, a ca-

racterística de ser mínima e estreita ajuda principalmente na redução do pessimismo gerado na computação dos intervalos. A propagação do pessimismo ao longo dos processos de otimização pode resultar em uma solução intervalar com largura tal que não ajuda no processo decisório. Dentre as funções teste apresentadas, a função de inclusão natural para ZDT3 não é estreita porque contém ocorrências repetidas de variáveis. Como consequência, o [I]RMOA I que depende da computação exaustiva de funções de inclusão teve seu desempenho prejudicado. No problema de otimização de sintonia dos controladores de processo a construção da função de inclusão das funções objetivo foi a tarefa mais complicada. Para sistemas de ordem maior que quatro, a construção dessas funções pelo método empregado não se aplica, pois as raízes do polinômio característico tem solução analítica até o quarto grau. A construção de funções de inclusão estreitas e a questão do pessimismo estão tratadas em Benhamou e Older [10], e Chabert e Jaulin [17].

Aplicação dos algoritmos propostos em um problema real estabeleceu uma conexão importante entre todas as etapas do processo de otimização. O problema escolhido foi a otimização de sintonia de controladores PID. Entre outras questões, foi detalhado como o problema deve ser modelado para ser tratado por métodos intervalares, com dedução analítica das funções de inclusão para os objetivos do problema específico. A estabilidade é, normalmente, uma característica desejável em sistemas de controle com malha fechada. Por isso, decidiu-se interpretar os critérios de estabilidade como restrições. Além disso, outras restrições relativas às singularidades dos métodos intervalares e ao desempenho do sistema de controle foram adicionadas. O resultado do tratamento das restrições foi apresentado graficamente na Fig.5.4. Analisando a figura, fica claro como os métodos intervalares excluem eficientemente as porções de espaço classificadas como não viáveis, segundo dada precisão. O estudo de caso foi finalizado com a aplicação dos algoritmos propostos. As fronteiras robustas foram comparadas, sendo os resultados dos algoritmos coerentes entre si. Este fato confirma que os métodos estão produzindo soluções corretas.

## 6.2 Trabalhos Futuros

A Otimização robusta multi-objetivo é um problema de custo computacionalmente caro por definição ( $\mathbf{f}(\mathbf{x}, \mathbf{p}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p}$ ). Por isso, boa parte das futuras direções de pesquisa foram relacionadas à otimização do tempo de busca pelas soluções robustas. Essas direções foram listadas a seguir.

Vários processos dos algoritmos apresentados são paralelizáveis, como a

bissecção de caixas por exemplo. Assim, esses processos podem tirar vantagens, em termos de tempo de execução, com uma nova codificação desenvolvida para processamento em paralelo.

Para aumentar a aplicabilidade dos algoritmos propostos, decidiu-se por não desenvolver métodos dependentes de cálculos de derivadas. No entanto, se o problema for contínuo, a idéia de se usar derivadas em intervalos acelerará convergência e aumentará a precisão do resultados.

Em todos os métodos propostos para otimização robusta, as caixas intervalares foram armazenadas em listas encadeadas. A lista encadeada mais utilizada neste trabalho é do tipo fila; a estrutura pilha foi empregada para executar a função CSC. No entanto, espera-se ganho em redução do esforço e com a organização se as filas forem substituídas por estruturas de dados baseadas em árvores.

A métrica MB[I] foi projetada e implementada para espaço dos objetivos com duas dimensões. A extensão dessa métrica para espaços maiores que 2D ampliará a aplicabilidade do método.

A técnica NB[I] também pode ser aplicada ao [I]RMOA I, por exemplo numa etapa após a geração de pontos da função grade. Quando surgisse um primeiro ponto pertencente a  $\mathbf{K}^+$  que dominasse todo o nicho, todos os pontos pertencentes aos nichos dominados seriam excluídos diretamente. Um procedimento similar pode ser usado quando se encontra pontos em  $\mathbf{K}^-$ . A expectativa é que o custo caia expressivamente.

Os nichos também poderiam ser mesclados ao algoritmo [I]RMOA II. A idéia seria não permitir que o método trabalhasse com um número de soluções candidatas acima de um valor máximo. Quando fosse ultrapassado esse limite a técnica de nicho seria empregada para eliminar soluções que ocupassem regiões mais concentradas. No entanto, o conceito de otimismo (2.8) para otimização robusta deverá ser utilizado para explicar a escolha de uma solução em detrimento de outra.

A técnica NB[I] foi guiada apenas por um parâmetro  $r_{bis}^{max}$  que determina o máximo de bissecções. No entanto, existem mais aspectos a investigar, por exemplo: a) qual a melhor maneira de reduzir problemas de pontos próximos classificados em nichos diferentes; b) qual o melhor valor  $r_{bis}^{max}$ ; e c) como seria o comportamento do algoritmo de bissecção caso o corte não passe pelo centro do eixo de maior largura.

A lista das sugestões para continuar pesquisa não termina nas sugestões acima. Entretanto, acredita-se que esses tópicos são pontos chave para o aprimoramento dos algoritmos propostos.

## BIBLIOGRAFIA

- [1] G. Alefeld. Inclusion methods for systems of nonlinear equations - the interval Newton method and modifications. In Jurgen Herzberger, editor, *Topics in Validated Computations*, pages 7–26. Elsevier, 1994.
- [2] G. Alefeld and G. Mayer. Interval analysis: theory and applications. *Journal of Computational and Applied Mathematics*, (121):421–464, 2000. Amsterdam.
- [3] B. D. O. Anderson, E. I. Jury, and Mansour. On robust hurwitz polynomials. *IEEE Transactions on Automatic Control*, 32(10):909–913, 1987.
- [4] J. Andersson. A survey of multiobjective optimization in engineering design, 2000.
- [5] K. J. Astrom and T. Hagglund. New tuning methods for pid controllers. In *European Control Conference*, pages 2456–2462, Rome, Italy, 1995.
- [6] V. Barichard and J. K. Hao. A population and interval constraint propagation algorithm for multi-objective optimization. In *Metaheuristics International Conference-MIC'03*, Kyoto, Japan, 2003.
- [7] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4), 1998.
- [8] A. Ben-Tal and A. Nemirovski. Robust optimization - methodology and applications. *Mathematical Programming (Series B)*, 92:453–480, 2002.
- [9] R. Benayoun, J. de Montgolfier, J. Tergny, and O. Laritchev. Linear programming with multiple objective functions: Step method (stem). *Mathematical Programming*, 1:366–375, 1971.
- [10] F. Benhamou and W. J. Older. Applying interval arithmetic to real, integer and Boolean constraints. *Logic Programming: The ALP Newsletter*, 6(2):13–14, 1993.
- [11] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98(1-3):49–71, 2003.
- [12] H. G. Beyer and B. Sendhoff. Robust optimization - a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196:3190–3218, 2007.
- [13] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2006.
- [14] G. C. Calafiore and M. C. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006.
- [15] C. Carreras and I. Walker. Interval methods for improved robot reliability estimation, 2000.

- 
- [16] L. G. Casado, I. García, J. A. Martínez, and Ya. D. Sergeyev. New interval analysis support functions using gradient information in a global minimization algorithm. *Journal of Global Optimization*, 25(4):345–362, 2003. Kluwer Academic Publishers.
- [17] G. Chabert and L. Jaulin. Computing the pessimism of inclusion functions. *Reliable computing*, 13(6):489–504, 2007.
- [18] A. Charnes and W. W. Cooper. Chance-constrained programming. *Management Science*, 6(1):73–79, 1959.
- [19] M. Chidambaram and R. P. Sree. A Simple Method of Tuning pid Controllers for Integration/Dead-Time Processes. *Computers and Chemical Engineering*, 27:211–215, 2003.
- [20] C. A. C. Coello and T. Pulido. Multiobjective optimization using micro-genetic algorithm. In L. S. et al., editor, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 274–282, California, 2001. Morgan Kaufmann Publishers.
- [21] C. A. C. Coello and M. R. Sierra. A coevolutionary multi-objective evolutionary algorithm. In *Congress on Evolutionary Computation - CEC'2003*, pages 482–489, Canderra, Australia, December 2003. IEEE Press.
- [22] G. H. Cohen and G. A. Coon. Theoretical consideration of retarded control. *Transactions of the ASME*, 75(1):827–834, 1953.
- [23] A. E. Csallner, T. Csendes, and M. Cs. Markót. Multisection in interval branch-and-bound methods for global optimization i. theoretical results. *J. Global Optimization*, (16):371–392, 2000.
- [24] P. Cztzak and A. Jaskiewicz. Pareto simulated annealing: a meta-heuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, 7(1):34–47, 1998.
- [25] M. Dahleh, A. Tesi, and A. Vicino. An overview of extremal properties for robust control of interval plants. *Automatica*, 29(3):707–721, 1993.
- [26] G. B. Dantzig. Linear programming under uncertainty. *Management Science*, 1(3-4):197–206, 1955.
- [27] M. Dao. *Projection des ensembles, application à l'estimation et à la commande robuste*. PhD thesis, Université d'Angers, 2006.
- [28] I. Das and J. E. Dennis. Normal-boundary intersection: a new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM J. Optim.*, (8):631–657, 1998.
- [29] M. Daumas. Past and future formalizations of iee 754, 854 and 754r standards, July 2002. Talk presented to the IEEE 754 committee.
- [30] M. D. Davis. *Game Theory, a nontechnical introduction*. Dover Publications, New York, 1983.
- [31] K. Deb. Multi-objective genetic algorithms: Problems difficulties and construction of test problems. *Evolutionary Computation Journal*, 7(3):205–230, 1999.
- [32] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, 2001.

- 
- [33] K. Deb and H. Gupta. Searching for robust pareto-optimal solutions in multi-objective optimization. In *EMO*, pages 150–164, 2005.
- [34] K. Deb, Agarwal S. Pratap, A., and T. Meyarivan. A fast and elist multiobjective genetic algorithms: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.
- [35] J. DeVale and P. J. Koopman Jr. Robust software - no more excuses. In *DSN '02: Proceedings of the 2002 International Conference on Dependable Systems and Networks*, pages 145–154. IEEE Computer Society, 2002.
- [36] L. C. Dias and J. N. Clímaco. On computing electre’s credibility indices under partial information. *Journal of Multi-Criteria Decision Analysis*, 8:74–92, 1999.
- [37] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, Cambridge, MA, 2004.
- [38] X. Du and W. Chen. Towards a better understanding of modeling feasibility robustness in engineering design, 1999.
- [39] L. El Ghaoui and G. Calafiori. Worst-case simulation of uncertain systems. In A. Tesi A. Garulli and A. Vicino, editors, *Robustness in Identification and Control*. Springer, 1999.
- [40] L. El Ghaoui, F. Oustry, and H. Lebret. Robust solutions to uncertain semidefinite programs. *SIAM J. OPTIM.*, 9(1):33–52, 1998.
- [41] R. A. Fisher. *Design of experiments*. Edinburgh: Oliver & Boyd, 1951.
- [42] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of 5th International Conference on Genetic Algorithm*, pages 416–423, California, 1993.
- [43] C. M. Fonseca and P. J. Fleming. An overview of evolutionary algorithms in multi-objective optimization. In *Evolutionary Computation Journal*, volume 3, pages 1–16, 1995.
- [44] Y. Gao, L. Shi, and P. Yao. Study on multi-objective genetic algorithm. In *Proceedings of 3rd World Congress on Intelligent Control and Automation*, Heifei, P. R. China, June 28-July 2 2000.
- [45] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.
- [46] D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of First International Conference on Genetic Algorithms and Their Applications*, pages 41–49, 1987.
- [47] J. J. Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 16(1):122–128, 1986.
- [48] E. R. Hansen. Interval arithmetic in matrix computations - part i. *SIAM Journal on Numerical Analysis*, (2):308–320, 1965.
- [49] E. R. Hansen. Global optimization using interval analysis - the one-dimensional case. *Journal of Optimization Theory and Applications*, (29):331–344, 1979.

- 
- [50] E. R. Hansen and G. W. Walster. *Global Optimization Using Interval Analysis*. Marcel Dekker, Inc, 2 edition, 2004.
- [51] M. P. Hansen and A. Jazskiewicz. Evaluating the quality of approximations to the non-dominated search. Technical report imm-rep-1998-7, Technical University of Denmark, March 1998.
- [52] R. J. Hanson. Interval arithmetic as a closed arithmetic system on a computer. Technical report, Jet Propulsion Laboratory, 1968.
- [53] B. Hayes. A lucid interval. *Computing Science*, 91(6):484–488, November-December 2003.
- [54] T. L. Heath. *On the measurement of the circle, in the book The Work of Archimedes*. Cambridge University Press, Cambridge, 1897.
- [55] D. Henrion and O. Bachelier. Low-order robust controller synthesis for interval plants. *International Journal of Control*, 74(1):1–9, 2001.
- [56] T. Hickey, Q. Ju, and M.H. Van Emden. Interval arithmetic: from principles to implementation. *Journal of ACM*, 48(5):1038–1068, September 2001.
- [57] J. H. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, 1975.
- [58] J. Horn, N. Nafpliotis, and D. E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Proc. of 1st ICEC*, pages 82–87, 1994.
- [59] C. C. Hsu and Yu C-Y. Design of optimal controller for interval plant from signal energy point of view via evolutionary approaches. *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 34(3):1609–1617, June 2004.
- [60] K. Ichida and Y. Fujii. An interval arithmetic method for global optimization. *Computing*, (23):85–97, 1979.
- [61] K. Ichida and Y. Fujii. Multicriterion optimization using interval analysis. *Computing*, (44):47–57, 1990.
- [62] L. Jaulin. Interval constraint propagation with application to bounded-error estimation. *Automatica*, (36):1547–1552, 2000.
- [63] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer, 2001.
- [64] L. Jaulin and E. Walter. Guaranteed tuning, with application to robust control and motion planning. *Automatica*, 32(8):1217–1221, 1996.
- [65] Y. Jin and J. Branke. Evolutionary optimization in uncertain environments-a survey. *Evolutionary Computation, IEEE Transactions on*, 9:303–317, 2005.
- [66] W. M. Kahan. A more complete interval arithmetic, 1968. Lecture notes for a summer course at the University of Michigan.
- [67] R. B. Kearfott. *Interval arithmetic methods for nonlinear systems and nonlinear optimization: an introduction review*, chapter Impact of Recent Computer Advances on Operations Research, pages 533–542. North-Holland, New York, NY, 1989.
- [68] R. B. Kearfott and V. Kreinovich. Applications of interval computations: An introduction. pages 1–22, 1996.

- 
- [69] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proc. in IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.
- [70] V. L. Kharitonov. Asymptotic stability of an equilibrium position of a family systems of linear differential equations. *Differential'nye Uraveniya*, 14(11):2086–2088, 1978.
- [71] H. Kitano. Biological robustness. *Nature Reviews - Genetics*, 5:826–837, 2004.
- [72] J. D. Knowles. *Local Search and Hybrid Evolutionary Algorithms for Pareto Optimization*. PhD thesis, Department of Computer Science, University of Reading, Reading, UK, 2002.
- [73] J. D. Knowles and D. W. Corne. Approximating the non-dominated front using pareto archived strategy. *Evolutionary Computation*, 8(2):149–172, 2000.
- [74] J. Koski and R. Silvennoinen. Norm methods and partial weighting in multicriterion optimization of structures. *Int. J. Numer. Methods Eng.*, (24):1101–1121, 1987.
- [75] P. Kouvelis and G. Yu. *Robust Discrete Optimization and Its Applications*. Kluwer Academic Publishers, 1997.
- [76] M. Laumanns, G. Rudolph, and H. P. Schwefel. A spatial predator-prey approach to multi-objective optimization: a preliminary study. In *Proceedings of the Fifth Conference on Parallel Problem Solving from Nature (PPSN V)*, volume 1498. Springer.
- [77] A. D. S. Lordelo, E. A. Juzzo, and P. A. V. Ferreira. Analysis and design of robust controllers using interval diophantine equation. *Reliable Computing*, 12:371–388, 2006.
- [78] S. K. Mahato and A. K. Bhunia. Interval-arithmetic-oriented interval computing technique for global optimization. *Applied Mathematics Research eXpress*, pages 1–19, 2006.
- [79] B. Mareschal. Weight stability intervals in multicriteria decision aid. *European Journal of Operational Research*, 33:54–64, 1988.
- [80] R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Struct Multidisc Optim*, 26:369–395, 2004.
- [81] A. Messac. Physical programming: effective optimization for computational design. *AIAA J.*, (34):149–158, 1996.
- [82] A. Messac, A. Ismail-Yahaya, and C. A. Mattson. The normalized normal constraint method for generating the pareto frontier. *Struct. Multidisc. Optim.*, (25):86–98, 2003.
- [83] A. Messac, C. Puemi-Sukam, and E. Melachrinoudis. Mathematical and pragmatic perspectives of physical programming. *AIAA J.*, (39):885–893, 2001.
- [84] K. Miettinen. *Nonlinear Multiobjective Optimization*, volume 12 of *International Series in Operations Research and Management Science*. Kluwer Academic Publishers, Dordrecht, 1999.
- [85] R. E. Moore. *Dimensional Analysis*. New Jersey: Prentice Hall, 1966.
- [86] R. E. Moore. On computing the range of a rational function of n variables over a bounded region. *Computing*, 15:1–15, 1976.

- 
- [87] R. E. Moore. *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, PA, 1979.
- [88] D. Morales and T. C. Son. Interval methods in robot navigation. *Reliable Computing*, 4(1):55–61, 1998.
- [89] T. Murata and H. Ishibuchi. Moga: Multi-objective genetic algorithms. In *Proc. of 2st IEEE ICEC*, pages 289–294, 1995.
- [90] A. Neumaier. Interval interaction for zeros of systems of equations. *BIT*, (25):256–273, 1985.
- [91] K. Ogata. *Engenharia de Controle Moderno*. Prentice Hall do Brasil Ltda, 1982.
- [92] A. Osyczka. An approach to multicriterion optimization problems for engineering design. *Comput. Methods Appl. Mech. Eng.*, (15):309–333, 1978.
- [93] A. Parkinson, C. Sorensen, and N. Pourhassan. A general approach for robust optimal design. "journal of mechanical design. *Transactions of the ASME*, 115(1):75–81, 1993.
- [94] R. O. Parreiras, J. H. R. D. Maciel, and J. A. Vasconcelos. The a posteriori decision in multiobjective optimization problems with smarts, promethee ii, and a fuzzy algorithm. *IEEE Transactions on Magnetics*, 42:1139–1142, 2006.
- [95] R. O. Parreiras and J. A. Vasconcelos. A multiplicative version of promethee ii applied to multiobjective optimization problems. *European Journal of Operational Research*, 183:729–740, 2007.
- [96] L. R. Pujara and A. Roy. On computing stabilizing controllers for siso interval plants. In *Proceedings of the American Control Conference*, volume 5, pages 3896–3901, Arlington, VA, USA, June 2001.
- [97] S. S. Rao and T. I. Freiheit. A modified game theory approach to multiobjective optimization. *J. Mech. Des.*, 113(286-291), 1991.
- [98] B. Roy. A missing link in OR-DA, robustness analysis. *Foundations of Computing and Decision Sciences*, 23(3):141–160, 1998.
- [99] G. Rudolph. Evolutionary search under partially ordered fitness sets. In *Proc. of International Symposium on Information Science Innovations in Enginnering of Natural and Artificial Intelligent Systems*, pages 818–822, 2001.
- [100] G. R. Ruetsch. An interval algorithm for multi-objective optimization. *Structural and Multidisciplinary Optimization*, 30(1):27–37, july 2005.
- [101] S. M. Rump. Interval computation with matlab. Nota de apresentação do toolbox INTLAB.
- [102] Y. Sawaragi, H. Nakayama, and T. Tanino. *Theory of Multiobjective Optimization*. Academic Press, Orlando, 1985.
- [103] J. D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithm. In *Proc. of 1st ICCA*, pages 93–100, 1985.
- [104] J. R. Schott. Fault tolerant design using single and multi-criteria genetic algorithms. Master's thesis, Department of Aeronautics and Aeronautics, Massachusetts Institute of Technology, Boston, MA, 1995.

- 
- [105] A. Sengupta and T. K. Pal. On comparing interval numbers. *European Journal of Operational Research*, 127(1):28–43, November 2000.
- [106] G. M. Siouris, G. Chen, and J. Wang. Tracking an incoming ballistic missile using an extended interval kalman filter. *IEEE Transactions on Aerospace and Electronic Systems*, 33(1):232–240, 1997.
- [107] S. Skelboe. Computation of rational interval functions. *BIT*, 14:87–95, 1974.
- [108] S. Skelboe. True worst-case analysis of linear electrical circuits by interval arithmetic. *IEEE Transactions on Circuits and Systems*, 26:874–879, 1979.
- [109] G. L. Soares. Algoritmos genéticos: Estudo, novas técnicas e aplicações. Master’s thesis, Universidade Federal de Minas Gerais, 1997.
- [110] G. L. Soares, R. L. Adriano, C. A. Maia, L. Jaulin, and J. A. Vasconcelos. Robust multi-objective team 22 problem: a case study of uncertainties in design optimization. In *Thirteenth Biennial IEEE Conference on Electromagnetic Field Computation CEFC 2008*, Athens, Greece, May 2008.
- [111] G. L. Soares, A. A. Boss, L. Jaulin, C. A. Maia, and J. A. Vasconcelos. An interval-based target tracking approach for range-only multistatic radar. *IEEE Transactions on Magnetics*, 44(6):1350–1353, 2008.
- [112] G. L. Soares, R. O. Parreiras, L. Jaulin, C. A. Maia, and J. A. Vasconcelos. Interval robust multi-objective algorithm. In *World Congress of Nonlinear Analysts WCNA 2008*, Orlando, Florida, July 2008.
- [113] G. L. Soares and J. A. Vasconcelos. Adaptação dinâmica de operadores em algoritmos genéticos. In *Segundo Congresso Brasileiro de Eletromagnetismo*, pages 155–158, Ouro Preto, novembro 1996.
- [114] G. L. Soares, J. A. Vasconcelos, and C. A. Maia. Algoritmo evolucionário por eleições. In *1º Seminário do Programa de Pós-Graduação em Engenharia Elétrica*, pages 1–5, Belo Horizonte, Brasil, 2005.
- [115] N. Srinivas and Deb K. Multiobjective function optimization using nondominated sorting genetic algorithms. *Evolutionary Computation*, 2(3):221–248, Fall 1995.
- [116] R. E. Steuer and R. U. Choo. An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26:326–344, 1983.
- [117] T. Sunaga. Theory of an interval algebra and its applications to numerical analysis. *RAAG Memoirs*, 2:547–564, 1958.
- [118] G. Taguchi. *Introduction to Quality Engineering*. American Supplier Institute, 1989.
- [119] R. H. C. Takahashi, P. L. D. Peres, and P. A. V. Ferreira. Multi-objective  $\mathcal{H}_2 / \mathcal{H}_\infty$  guaranteed cost pid design. *IEEE Control Systems*, pages 37–47, October 1997.
- [120] G. L. R. Vaccaro. *Solução de Equações Intervalares*. PhD thesis, Universidade Federal do Rio Grande do Sul, 2001.
- [121] D. V. Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analysis, and New Innovations*. PhD thesis, Air Force Institute of Technology, Dayton, OH, 1999.

- 
- [122] A. Visioli. Optimal Tuning of PID Controllers for Integral and Unstable Processer. *IEEE Proc - Theory Appl.*, 148(2):180–184, 2001.
- [123] G. Walster, J. D. Pryce, and E. R. Hansen. Practical, exception-free interval arithmetic on extended reals. *SIAM Journal on Scientific Computing*. The paper was provisionally accepted for publication; has since been significantly revised; and will be resubmitted for publication.
- [124] F. M. Waltz. An engineering approach: hierarchical optimization criteria. *IEEE Transaction Autom. Control*, AC-12:179–180, 1967.
- [125] Y.-G. Wang and W.-J. Cai. Advanced proportional-integral-derivative tuning for integrating and unstable processes with gain and phase margin specifications. *American Chemical Society*, 41:2910–2914, 2002.
- [126] P. L. Yu. Cone convexity, cone extreme points, and nondominated solutions in decision problems with multiobjectives. *J. Optim. Theory Appl.*, (14):319–377, 1974.
- [127] P. L. Yu. *Multiple-Criteria Optimization: Techniques and Extensions*. Plenum Press, New York, 1985.
- [128] M. Zeleny. *Multiple Criteria Decision Making*. New York: McGraw Hill, 1982.
- [129] J. G. Ziegler and N. B. Nichols. Optimum setting for automatic controllers. *Transactions of the ASME*, 64:759–768, 1942.
- [130] S. Zionts. Multiple criteria mathematical programming: an updated overview and several approaches. *Mitra, G. (ed.) Mathematical Models for Decision Support*, pages 135–167, Berlin: Springer-Verlag 1988.
- [131] E. Zitzler. *Evolutionary Algorithms or Multiobjective Optimization: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology, Zurich, Switzerland, 1999.
- [132] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation Journal*, 8(2):125–148, 2000.
- [133] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms - a comparative case study and the strength pareto approach. In *Parallel Problem Solving from Nature V (PPSN-V)*, pages 292–301, 1998.

## APÊNDICE

## A. EQUAÇÕES PARA OS CÁLCULOS DO TEMPO DE PICO E DO SOBRE-SINAL MÁXIMO

Considere a resposta do sistema

$$z(t) = 1 + e^{-\frac{(1+K_d)}{2}t} \left( -\cosh\left(\frac{\sqrt{\alpha}}{2}t\right) + \frac{(-1+K_d)}{\sqrt{\alpha}} \sinh\left(\frac{\sqrt{\alpha}}{2}t\right) \right) \quad (\text{A.1})$$

mas,

$$\cosh\left(\frac{\sqrt{\alpha}}{2}t\right) = \frac{e^{\frac{\sqrt{\alpha}}{2}t} + e^{-\frac{\sqrt{\alpha}}{2}t}}{2} \quad \text{e} \quad \sinh\left(\frac{\sqrt{\alpha}}{2}t\right) = \frac{e^{\frac{\sqrt{\alpha}}{2}t} - e^{-\frac{\sqrt{\alpha}}{2}t}}{2} \quad (\text{A.2})$$

realizando as substituições e multiplicando toda equação por 2, fica,

$$z(t) = 1 + e^{-\frac{1+K_d}{2}t} \left( -e^{\frac{\sqrt{\alpha}}{2}t} - e^{-\frac{\sqrt{\alpha}}{2}t} + \frac{-1+K_d}{\sqrt{\alpha}} \left( e^{\frac{\sqrt{\alpha}}{2}t} - e^{-\frac{\sqrt{\alpha}}{2}t} \right) \right) \quad (\text{A.3})$$

$$z(t) = 1 + e^{-\frac{1+K_d}{2}t} \left( -e^{\frac{\sqrt{\alpha}}{2}t} \left( 1 + \frac{1-K_d}{\sqrt{\alpha}} \right) - e^{-\frac{\sqrt{\alpha}}{2}t} \left( 1 + \frac{-1+K_d}{\sqrt{\alpha}} \right) \right) \quad (\text{A.4})$$

$$z(t) = 1 - \left( 1 + \frac{1-K_d}{\sqrt{\alpha}} \right) e^{\frac{\sqrt{\alpha}-1-K_d}{2}t} - \left( 1 + \frac{-1+K_d}{\sqrt{\alpha}} \right) e^{\frac{-\sqrt{\alpha}-1-K_d}{2}t} \quad (\text{A.5})$$

O tempo de pico ocorre no primeiro máximo da resposta após a referência. Logo, obtém-se  $t_p$  derivando-se (A.5) e igualando-se a expressão resultante zero. Assim,

$$\begin{aligned} \frac{dz(t)}{dt} &= 0 - \left( 1 + \frac{1-K_d}{\sqrt{\alpha}} \right) \left( \frac{\sqrt{\alpha}-1-K_d}{2} \right) e^{\frac{\sqrt{\alpha}-1-K_d}{2}t_p} \dots \\ &\dots - \left( 1 + \frac{-1+K_d}{\sqrt{\alpha}} \right) \left( \frac{-\sqrt{\alpha}-1-K_d}{2} \right) e^{\frac{-\sqrt{\alpha}-1-K_d}{2}t_p} \end{aligned} \quad (\text{A.6})$$

$$\begin{aligned} 0 &= -(\sqrt{\alpha}+1-K_d)(\sqrt{\alpha}-1-K_d)e^{\frac{\sqrt{\alpha}-1-K_d}{2}t_p} \dots \\ &\dots - (\sqrt{\alpha}-1+K_d)(-\sqrt{\alpha}-1-K_d)e^{\frac{-\sqrt{\alpha}-1-K_d}{2}t_p} \end{aligned} \quad (\text{A.7})$$

após rearranjo, fica,

$$\frac{e^{\frac{\sqrt{\alpha}-1-K_d}{2}t_p}}{e^{\frac{-\sqrt{\alpha}-1-K_d}{2}t_p}} = -\frac{(\sqrt{\alpha}-1+K_d)(-\sqrt{\alpha}-1-K_d)}{(\sqrt{\alpha}+1-K_d)(\sqrt{\alpha}-1-K_d)} \quad (\text{A.8})$$

$$e^{\sqrt{\alpha}t_p} = \frac{1 - (\sqrt{\alpha} + K_d)^2}{1 - (\sqrt{\alpha} - K_d)^2} \quad (\text{A.9})$$

e finalmente,

$$t_p = \frac{1}{\sqrt{\alpha}} \ln \left( \frac{1 - (\sqrt{\alpha} + K_d)^2}{1 - (\sqrt{\alpha} - K_d)^2} \right) \quad \text{e} \quad M_p = z(t_s) - 1. \quad (\text{A.10})$$

## B. EQUAÇÃO PARA O TEMPO DE SUBIDA

Considere a resposta do sistema (5.18)

$$z(t) = 1 + e^{-\frac{(1+K_d)}{2}t} \left( -\cosh\left(\frac{\sqrt{\alpha}}{2}t\right) + \frac{(-1+K_d)}{\sqrt{\alpha}} \sinh\left(\frac{\sqrt{\alpha}}{2}t\right) \right) \quad (\text{B.1})$$

Segundo a definição, no tempo final da subida a resposta atinge o valor da referência, ou seja  $z(t_s) = 1$ , logo

$$1 = 1 + e^{-\frac{(1+K_d)}{2}t_s} \left( -\cosh\left(\frac{\sqrt{\alpha}}{2}t_s\right) + \frac{(-1+K_d)}{\sqrt{\alpha}} \sinh\left(\frac{\sqrt{\alpha}}{2}t_s\right) \right) \quad (\text{B.2})$$

$$0 = e^{-\frac{(1+K_d)}{2}t_s} \left( -\cosh\left(\frac{\sqrt{\alpha}}{2}t_s\right) + \frac{(-1+K_d)}{\sqrt{\alpha}} \sinh\left(\frac{\sqrt{\alpha}}{2}t_s\right) \right) \quad (\text{B.3})$$

como  $\frac{1}{2}e^{-\frac{(1+K_d)}{2}t_s} \neq 0$ , então

$$0 = -\cosh\left(\frac{\sqrt{\alpha}}{2}t_s\right) + \frac{(-1+K_d)}{\sqrt{\alpha}} \sinh\left(\frac{\sqrt{\alpha}}{2}t_s\right) \quad (\text{B.4})$$

mas,

$$\cosh\left(\frac{\sqrt{\alpha}}{2}t\right) = \frac{e^{\frac{\sqrt{\alpha}}{2}t} + e^{-\frac{\sqrt{\alpha}}{2}t}}{2} \quad \text{e} \quad \sinh\left(\frac{\sqrt{\alpha}}{2}t\right) = \frac{e^{\frac{\sqrt{\alpha}}{2}t} - e^{-\frac{\sqrt{\alpha}}{2}t}}{2} \quad (\text{B.5})$$

realizando as substituições e multiplicando toda equação por 2, fica,

$$0 = -e^{\frac{\sqrt{\alpha}}{2}t} - e^{-\frac{\sqrt{\alpha}}{2}t} + \frac{(-1+K_d)}{\sqrt{\alpha}} \left( e^{\frac{\sqrt{\alpha}}{2}t} - e^{-\frac{\sqrt{\alpha}}{2}t} \right) \quad (\text{B.6})$$

$$0 = -e^{\frac{\sqrt{\alpha}}{2}t} \left( 1 - \frac{(-1+K_d)}{\sqrt{\alpha}} \right) + e^{-\frac{\sqrt{\alpha}}{2}t} \left( -1 - \frac{(-1+K_d)}{\sqrt{\alpha}} \right) \quad (\text{B.7})$$

multiplica-se toda equação por  $\sqrt{\alpha}$ , e após a organização fica

$$e^{\frac{\sqrt{\alpha}}{2}t_s} (\sqrt{\alpha} + 1 - K_d) = e^{-\frac{\sqrt{\alpha}}{2}t_s} (-\sqrt{\alpha} + 1 - K_d) \quad (\text{B.8})$$

$$e^{\sqrt{\alpha}t_s} = \frac{-\sqrt{\alpha} + 1 - K_d}{\sqrt{\alpha} + 1 - K_d} \quad (\text{B.9})$$

e finalmente,

$$t_s = \frac{1}{\sqrt{\alpha}} \ln \left( \frac{-\sqrt{\alpha} + 1 - K_d}{\sqrt{\alpha} + 1 - K_d} \right) \quad (\text{B.10})$$

## C. EQUAÇÕES PARA O ISE

Considere a definição do erro integral quadrático

$$\text{ISE} = \int_0^{\infty} e^2(t) dt. \quad (\text{C.1})$$

mas o sinal de erro é definido como  $e(t) = r(t) - z(t)$  ou mais precisamente,

$$e(t) = 1 - z(t), \quad (\text{C.2})$$

pois  $r(t) = 1$ ,  $t > 0$ . Assim, considerando a resposta do sistema

$$z(t) = 1 + e^{-\frac{(1+K_d)}{2}t} \left( -\cosh\left(\frac{\sqrt{\alpha}}{2}t\right) + \frac{(-1+K_d)}{\sqrt{\alpha}} \sinh\left(\frac{\sqrt{\alpha}}{2}t\right) \right), \quad (\text{C.3})$$

substituindo (C.3) em (C.2), tem-se

$$e(t) = e^{-\frac{(1+K_d)}{2}t} \left( -\cosh\left(\frac{\sqrt{\alpha}}{2}t\right) + \frac{(-1+K_d)}{\sqrt{\alpha}} \sinh\left(\frac{\sqrt{\alpha}}{2}t\right) \right). \quad (\text{C.4})$$

mas,

$$\cosh\left(\frac{\sqrt{\alpha}}{2}t\right) = \frac{e^{\frac{\sqrt{\alpha}}{2}t} + e^{-\frac{\sqrt{\alpha}}{2}t}}{2} \quad \text{e} \quad \sinh\left(\frac{\sqrt{\alpha}}{2}t\right) = \frac{e^{\frac{\sqrt{\alpha}}{2}t} - e^{-\frac{\sqrt{\alpha}}{2}t}}{2} \quad (\text{C.5})$$

realizando as substituições e multiplicando toda equação por 2, fica,

$$e(t) = -e^{-\frac{1+K_d}{2}t} \left( -e^{\frac{\sqrt{\alpha}}{2}t} - e^{-\frac{\sqrt{\alpha}}{2}t} + \frac{-1+K_d}{\sqrt{\alpha}} \left( e^{\frac{\sqrt{\alpha}}{2}t} - e^{-\frac{\sqrt{\alpha}}{2}t} \right) \right) \quad (\text{C.6})$$

$$e(t) = -e^{-\frac{1+K_d}{2}t} \left( -e^{\frac{\sqrt{\alpha}}{2}t} \left( 1 + \frac{1-K_d}{\sqrt{\alpha}} \right) - e^{-\frac{\sqrt{\alpha}}{2}t} \left( 1 + \frac{-1+K_d}{\sqrt{\alpha}} \right) \right) \quad (\text{C.7})$$

$$e(t) = \left( 1 + \frac{1-K_d}{\sqrt{\alpha}} \right) e^{\frac{\sqrt{\alpha}-1-K_d}{2}t} + \left( 1 + \frac{-1+K_d}{\sqrt{\alpha}} \right) e^{\frac{-\sqrt{\alpha}-1-K_d}{2}t} \quad (\text{C.8})$$

Considere as notações  $A = \left( 1 + \frac{1-K_d}{\sqrt{\alpha}} \right)$ ,  $B = \left( 1 + \frac{-1+K_d}{\sqrt{\alpha}} \right)$ ,  $\lambda = \frac{\sqrt{\alpha}-1-K_d}{2}$  e  $\mu = \frac{-\sqrt{\alpha}-1-K_d}{2}$ . A equação (C.8) é reescrita como

$$e(t) = Ae^{\lambda t} + Be^{\mu t}. \quad (\text{C.9})$$

Elevando o sinal de erro ao quadrado, fica

$$e^2(t) = A^2 e^{2\lambda t} + 2AB e^{(\lambda+\mu)t} + B^2 e^{2\mu t}. \quad (\text{C.10})$$

Substituindo o resultado de (C.10) em (C.1), fica

$$\text{ISE} = A^2 \int_0^\infty e^{2\lambda t} dt + 2AB \int_0^\infty e^{(\lambda+\mu)t} dt + B^2 \int_0^\infty e^{2\mu t} dt, \quad (\text{C.11})$$

e após a integração, tem-se

$$\text{ISE} = \frac{A^2}{2\lambda} [e^{2\lambda t}]_0^\infty + \frac{2AB}{\lambda + \mu} [e^{(\lambda+\mu)t}]_0^\infty + \frac{B^2}{2\mu} [e^{2\mu t}]_0^\infty. \quad (\text{C.12})$$

A dedução prossegue com a análise dos termos de (C.12) quando  $t \rightarrow 0$  e quando  $t \rightarrow \infty$ . Os expoentes dos termos contém apenas  $\lambda$  e  $\mu$ . A variável  $\mu$  é sempre negativa: a)  $K_p$  é não negativo pelo espaço em que foi definido; b)  $\sqrt{\alpha'} > 1$  segundo a restrição  $g_3$  da formulação robusta (5.22). Similarmente, a variável  $\lambda$  é sempre negativa pelo atendimento da restrição  $g_4$  da formulação robusta (5.22). Assim, sendo  $\lambda < 0$  e  $\mu < 0$  tem-se

$$\lim_{t \rightarrow \infty} e^{2\lambda t} = 0, \quad \lim_{t \rightarrow \infty} e^{(\lambda+\mu)t} = 0 \quad \text{e} \quad \lim_{t \rightarrow \infty} e^{2\mu t} = 0; \quad (\text{C.13})$$

$$\lim_{t \rightarrow 0} e^{2\lambda t} = 1, \quad \lim_{t \rightarrow 0} e^{(\lambda+\mu)t} = 1 \quad \text{e} \quad \lim_{t \rightarrow 0} e^{2\mu t} = 1. \quad (\text{C.14})$$

Os resultados (C.13) e (C.14) em (C.12) resulta

$$\text{ISE} = \frac{A^2}{2\lambda} [0 - 1] + \frac{2AB}{\lambda + \mu} [0 - 1] + \frac{B^2}{2\mu} [0 - 1], \quad (\text{C.15})$$

e portanto

$$\text{ISE} = -\frac{A^2}{2\lambda} - \frac{2AB}{\lambda + \mu} - \frac{B^2}{2\mu}. \quad (\text{C.16})$$

Substituindo as notações tem-se

$$\text{ISE} = -\frac{\left(1 + \frac{1-K_d}{\sqrt{\alpha}}\right)^2}{2\frac{\sqrt{\alpha}-1-K_d}{2}} - \frac{2\left(1 + \frac{1-K_d}{\sqrt{\alpha}}\right)\left(1 + \frac{-1+K_d}{\sqrt{\alpha}}\right)}{\frac{\sqrt{\alpha}-1-K_d}{2} + \frac{-\sqrt{\alpha}-1-K_d}{2}} - \frac{\left(1 + \frac{-1+K_d}{\sqrt{\alpha}}\right)^2}{2\frac{-\sqrt{\alpha}-1-K_d}{2}}, \quad (\text{C.17})$$

$$\text{ISE} = -\frac{\left(1 + \frac{1-K_d}{\sqrt{\alpha}}\right)^2}{\sqrt{\alpha}-1-K_d} - \frac{2\left(1 + \frac{1-K_d}{\sqrt{\alpha}}\right)\left(1 + \frac{-1+K_d}{\sqrt{\alpha}}\right)}{-1-K_d} - \frac{\left(1 + \frac{-1+K_d}{\sqrt{\alpha}}\right)^2}{-\sqrt{\alpha}-1-K_d}, \quad (\text{C.18})$$

$$\text{ISE} = -\frac{\left(1 + \frac{1-K_d}{\sqrt{\alpha}}\right)^2}{\sqrt{\alpha}-1-K_d} - \frac{2\left(1 + \frac{1-K_d}{\sqrt{\alpha}}\right)\left(1 - \frac{1-K_d}{\sqrt{\alpha}}\right)}{-1-K_d} - \frac{\left(1 - \frac{1-K_d}{\sqrt{\alpha}}\right)^2}{-\sqrt{\alpha}-1-K_d}. \quad (\text{C.19})$$

Finalmente,

$$\text{ISE} = -\frac{\left(1 + \frac{1-K_d}{\sqrt{\alpha}}\right)^2}{\sqrt{\alpha}-1-K_d} - \frac{2\left(1 - \left(\frac{1-K_d}{\sqrt{\alpha}}\right)^2\right)}{-1-K_d} - \frac{\left(1 - \frac{1-K_d}{\sqrt{\alpha}}\right)^2}{-\sqrt{\alpha}-1-K_d}. \quad (\text{C.20})$$

Outras simplificações podem ser possíveis. No entanto, a equação já atende a finalidade de implementação.