

**FERNANDO RICARDO MOREIRA**

**OTIMIZAÇÃO ROBUSTA  
MULTIOBJETIVO PARA O PROJETO  
DE SISTEMAS EM ENGENHARIA**



**UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE ENGENHARIA MECÂNICA  
2015**

**FERNANDO RICARDO MOREIRA**

**OTIMIZAÇÃO ROBUSTA  
MULTIOBJETIVO PARA O PROJETO  
DE SISTEMAS EM ENGENHARIA**

**Tese** apresentada ao Programa de Pós-graduação em Engenharia Mecânica da Universidade Federal de Uberlândia, como parte dos requisitos para a obtenção do título de **DOCTOR EM ENGENHARIA MECÂNICA**.

Área de concentração: Mecânica dos Sólidos e Vibrações.

Orientador: Prof. Dr. Valder Steffen Jr.

Co-orientador: Prof. Dr. Fran Sérgio Lobato

Uberlândia - MG

2015

Dados Internacionais de Catalogação na Publicação (CIP)  
Sistema de Bibliotecas da UFU, MG, Brasil.

---

M838o  
2015

Moreira, Fernando Ricardo, 1982-  
Otimização robusta multiobjetivo para o projeto de sistemas em engenharia / Fernando Ricardo Moreira. - 2015.  
267 f. : il.

Orientador: Valder Stefen Junior.  
Coorientador: Fran Sérgio Lobato.  
Tese (doutorado) - Universidade Federal de Uberlândia, Programa de Pós-Graduação em Engenharia Mecânica.  
Inclui bibliografia.

1. Engenharia mecânica - Teses. 2. Otimização matemática - Teses.  
I. Stefen Junior, Valder. II. Lobato, Fran Sérgio, 1976- III. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Engenharia Mecânica. IV. Título.

*Dedico este trabalho a minha família, em especial a minha amada esposa Samantha e aos nossos filhos Pedro e Mariana.*

# Agradecimentos

Ao meu Deus, pois tem sido fiel e cumprido em mim suas promessas.

À Universidade Federal de Uberlândia e à Faculdade de Engenharia Mecânica pela oportunidade de realizar meu doutorado em um programa de nível de excelência internacional.

Ao Prof. Dr. Valder Steffen Jr. pela orientação e pela sensibilidade na escolha do tema da minha pesquisa. Não poderia me esquecer de agradecer-lo pela maneira tão especial que o senhor me tratou nos momentos em que eu mais precisei de ajuda. Meus mais sinceros agradecimentos.

Ao Prof. Dr. Fran Sérgio Lobato que me co-orientou e me ajudou enormemente na parte computacional.

Ao grande amor da minha vida, minha esposa Samantha. Sua ajuda nos momentos difíceis foi fundamental para que eu pudesse suportá-los.

Aos meus filhos Pedro e Mariana. Vocês são o motivo da minha perseverança e da minha alegria.

Aos meus pais, irmãs e sobrinhas que sempre acreditaram e tiveram bastante orgulho de mim.

Aos meus amigos e colegas de trabalho na UFG/Jataí. Obrigado pelo apoio nestes últimos anos.

À Igreja Assembléia de Deus, Ministério Vila Nova em Jataí pelo amparo espiritual.

MOREIRA, F. R., **Otimização Robusta Multiobjetivo para o Projeto de Sistemas em Engenharia**. 2015. Tese de Doutorado, Universidade Federal de Uberlândia.

## Resumo

O objetivo deste trabalho foi o desenvolvimento de um algoritmo de otimização multiobjetivo robusto usando como base o Algoritmo de Colônia de Vagalumes associado ao conceito de Média Efetiva. Os principais operadores para extensão do algoritmo para o caso multiobjetivo foram o ordenamento das curvas de Pareto por meio de *rank* e o truncamento de soluções por meio do operador distância de multidão. Para inserir a robustez ao processo de otimização foi usado a definição de média efetiva em detrimento ao uso comum na literatura das medidas de expectância. A metodologia proposta foi testada em problemas matemáticos cujas curvas de Pareto nominais e robustas eram conhecidas de modo a avaliar a qualidade do algoritmo proposto via métricas de convergência e de diversidade. O Algoritmo proposto mostrou-se bastante eficiente em suas tarefas de convergência e garantia de diversidade das soluções. A metodologia também foi aplicada para o projetos de sistemas mecânicos clássicos, bem como para o projeto de um rotor flexível com respeito ao posicionamento de suas velocidades críticas. As principais contribuições desta tese foram o desenvolvimento de uma ferramenta computacional para tratamento de problemas de otimização multiobjetivo robusta, a análise e interpretação da influência do parâmetro de robustez sobre as curvas de Pareto robustas em comparação com as curvas nominais e a formação de um banco de dados para futuras comparações com outras estratégias de otimização multiobjetivo robusta. Os resultados indicam que a abordagem proposta surge como uma estratégia interessante para o projeto robusto de sistemas de engenharia.

*Palavras Chave: Otimização Robusta Multiobjetivo, Curva de Pareto, Algoritmo de Colônia de Vagalumes, Média Efetiva.*

MOREIRA, F. R., **Robust Optimization Multiobjective for Engineering System Design**, 2015. Ph.D. Thesis, Universidade Federal de Uberlândia.

## Abstract

The aim of this work was the development of a robust multiobjective optimization algorithm by using as a reference the Fireflies Colony Algorithm associated with the concept of Effective Mean. The main operators for the extension of the algorithm for multiobjective case were the ordering of Pareto curves by means of ranking procedures and the truncation of solutions through the crowding distance operator. To insert robustness to the optimization process, the mean effective definition was used instead the commonly used expectancy measures, as suggested by the literature. The proposed methodology was tested in mathematical problems whose nominal and robust Pareto curves were known. In addition, so as to evaluate the quality of the algorithm proposed metrics of convergence and diversity were taken into account. The proposed algorithm proved to be very efficient with respect to convergence and diversity of solutions. The methodology was also applied to design classical mechanical systems, including the design a flexible rotor with respect to the position of the critical speeds. The main contributions of this thesis was the development of a computational tool for the treatment of robust multi-objective optimization problems, the analysis and interpretation of the influence of robustness parameter on robust Pareto curves compared with the nominal curves and the formation of a bank data for future comparisons with other robust multi-objective optimization strategies. The results indicate that the approach proposed arises as an interesting strategy for the robust design of engineering systems.

*Keywords: Robust Optimization Multiobjective, Pareto curve, Firefly Colony Algorithm, Mean Effective.*

# Lista de Figuras

2.1	Formuladores do conceito de ótimo multiobjetivo . . . . .	14
2.2	Opções de escolha de soluções em uma situação hipotética. . . . .	14
2.3	Curvas de Pareto para problemas com dois objetivos . . . . .	19
2.4	Geometria do espaço de objetivos com curva de Pareto local . . . . .	19
2.5	Metas em otimização multiobjetivo: convergência e diversidade . . . . .	20
3.1	Fluxograma de passos para resolução de POMO sem o uso do critério de dominância de Pareto . . . . .	29
3.2	Método da soma ponderada: casos convexo e não convexo . . . . .	33
3.3	Método $\xi$ -restrito para um problema com dois objetivos . . . . .	36
3.4	Diferenças entre métricas no método do critério global . . . . .	41
4.1	Aplicação de uma iteração do método da Bissecção . . . . .	56
4.2	Escolha de um ponto $z$ no intervalo $[a, b]$ . . . . .	57
4.3	Aplicação de uma iteração no método da Seção Áurea . . . . .	58
4.4	Conjunto de Descida $\mathcal{D}_f(x)$ . . . . .	62
4.5	Geometria do método do Gradiente com passo obtido pela minimização exata . . . . .	64
4.6	Geometria de aplicação do método de Newton . . . . .	71
5.1	Fluxograma esquemático do processo de otimização de um AG . . . . .	82
5.2	Representação de um operador de cruzamento pontual . . . . .	84
5.3	Representação de um operador de mutação em dois pontos . . . . .	85
5.4	Geração de um vetor doador no algoritmo ED . . . . .	89
5.5	Cruzamento binomial no algoritmo ED . . . . .	90
5.6	Movimento de uma partícula no algoritmo EP . . . . .	97
5.7	<i>Ranking</i> de soluções obtido pelo critério de dominância . . . . .	103
5.8	Soluções agrupadas em nichos . . . . .	104
5.9	Representação do operador distância de multidão do NSGA II . . . . .	107
5.10	Representação do mecanismo de corte do NSGA II . . . . .	107
5.11	Situação de erro no mecanismo de corte por distância de multidão . . . . .	108
6.1	Ótimo global versus ótimo robusto . . . . .	112
6.2	Principais fontes de incertezas em projetos . . . . .	117



6.3	Amostragem pelos métodos de Monte Carlo e Hipercubo Latino. . . . .	126
6.4	Sensibilidade de soluções em um espaço com dois objetivos . . . . .	128
7.1	Ordenamento por <i>rank</i> realizado pelo algoritmo NSGA II . . . . .	136
7.2	Análise da violação de restrições para o caso robusto devido a perturbação de uma solução viável. . . . .	142
7.3	Arquivos componentes do algoritmo MOFA. . . . .	147
8.1	Curva de Pareto nominal e obtida pelo MOFA para o problema teste 1. . .	153
8.2	Curvas de Pareto robustas para o problema teste 1. . . . .	154
8.3	Curvas de Pareto robustas para o problema Teste 2. . . . .	156
8.4	Curva de Pareto nominais (global e local) e obtida pelo MOFA para o problema teste 3. . . . .	158
8.5	Curvas de Pareto robustas para o problema teste 3. . . . .	160
8.6	Curva de Pareto nominal e obtida pelo algoritmo MOFA para o problema teste 4. . . . .	162
8.7	Curvas de Pareto robustas para o problema teste 4. . . . .	163
8.8	Curva de Pareto nominal e obtida pelo algoritmo MOFA para o problema teste 5. . . . .	165
8.9	Curvas de Pareto robustas para o problema teste 5. . . . .	167
8.10	Curva de Pareto nominal e obtida pelo algoritmo MOFA para o problema teste 6. . . . .	168
8.11	Curvas de Pareto robustas para o problema teste 6. . . . .	170
8.12	Curvas de Pareto robustas para o problema teste 7. . . . .	171
8.13	Sensibilidade do algoritmo MOFA. Métrica de convergência <i>versus</i> variação de $\beta_0$ . . . . .	173
8.14	Sensibilidade do algoritmo MOFA. Métrica de convergência <i>versus</i> variação de $\gamma$ . . . . .	174
8.15	Sensibilidade do algoritmo MOFA. Métrica de convergência <i>versus</i> variação de $N_{amt}$ . . . . .	175
8.16	Sensibilidade do algoritmo MOFA. Métrica de diversidade <i>versus</i> variação de $\beta_0$ . . . . .	176
8.17	Sensibilidade do algoritmo MOFA. Métrica de diversidade <i>versus</i> variação de $\gamma$ . . . . .	176
8.18	Sensibilidade do algoritmo MOFA. Métrica de diversidade <i>versus</i> variação de $N_{amt}$ . . . . .	177
9.1	Geometria do problema das duas barras . . . . .	183
9.2	Curvas de Pareto robustas e nominal para o problema de duas barras. . .	184
9.3	Valores das variáveis para os casos nominal e robusto para os parâmetros de robustez $\delta = 0,00075$ e $\delta = 0,00125$ . . . . .	186

9.4	Valores das variáveis para os casos nominal e robusto para os parâmetros de robustez $\delta = 0,00075$ e $\delta = 0,00125$ . . . . .	186
9.5	Valores das variáveis para os casos nominal e robusto para os parâmetros de robustez $\delta = 0,00075$ e $\delta = 0,00125$ . . . . .	187
9.6	Geometria do problema da viga em balanço . . . . .	187
9.7	Curvas de Pareto robustas e nominal para o problema da viga em balanço obtidas pelo MOFA. . . . .	189
9.8	Valores das variáveis para os casos nominal e robusto para os parâmetros de robustez $\delta = 0,0009$ e $\delta = 0,0013$ . . . . .	190
9.9	Geometria do problema da viga soldada . . . . .	191
9.10	Curvas de Pareto nominal e robustas para o problema da viga soldada. . .	193
9.11	Variáveis $h$ e $l$ para os casos nominal e robusto ( $\delta = 0,10, \delta = 0,15$ ). . . . .	195
9.12	Variáveis $h$ e $t$ para os casos nominal e robusto ( $\delta = 0,10, \delta = 0,15$ ). . . . .	195
9.13	Variáveis $h$ e $b$ para os casos nominal e robusto ( $\delta = 0,10, \delta = 0,15$ ). . . . .	196
9.14	Variáveis $l$ e $t$ para os casos nominal e robusto ( $\delta = 0,10, \delta = 0,15$ ). . . . .	196
9.15	Variáveis $l$ e $b$ para os casos nominal e robusto ( $\delta = 0,10, \delta = 0,15$ ). . . . .	197
9.16	Variáveis $t$ e $b$ para os casos nominal e robusto ( $\delta = 0,10, \delta = 0,15$ ). . . . .	197
9.17	Etapas da resolução de problemas reais utilizando metamodelos . . . . .	200
9.18	Robô industrial com um mecanismo de mola hidráulica . . . . .	202
9.19	Curva de Pareto nominal para o projeto ótimo de um robô industrial. . . . .	206
9.20	Curva de Pareto robusta ( $\delta = 0,1$ ) para o projeto ótimo de um robô industrial. . . . .	206
9.21	Curva de Pareto robusta ( $\delta = 0,2$ ) para o projeto ótimo de um robô industrial. . . . .	207
9.22	$y_1^{met} \times y_2^{met}$ para o problema do projeto ótimo robusto ( $\delta = 0,1$ ) de um robô industrial. . . . .	207
9.23	$y_1^{met} \times y_3^{met}$ para o problema do projeto ótimo robusto ( $\delta = 0,1$ ) de um robô industrial. . . . .	208
9.24	$y_2^{met} \times y_3^{met}$ para o problema do projeto ótimo robusto ( $\delta = 0,1$ ) de um robô industrial. . . . .	208
9.25	$y_1^{met} \times y_2^{met}$ para o problema do projeto ótimo robusto ( $\delta = 0,2$ ) de um robô industrial. . . . .	209
9.26	$y_1^{met} \times y_3^{met}$ para o problema do projeto ótimo robusto ( $\delta = 0,2$ ) de um robô industrial. . . . .	209
9.27	$y_2^{met} \times y_3^{met}$ para o problema do projeto ótimo robusto ( $\delta = 0,2$ ) de um robô industrial. . . . .	210
9.28	Relação $x_1 \times x_2$ para o problema do projeto ótimo robusto de um robô industrial. . . . .	212
9.29	Relação $x_1 \times x_3$ para o problema do projeto ótimo robusto de um robô industrial. . . . .	212
9.30	Relação $x_1 \times x_4$ para o problema do projeto ótimo robusto de um robô industrial. . . . .	213

9.31	Relação $x_2 \times x_3$ para o problema do projeto ótimo robusto de um robô industrial. . . . .	213
9.32	Relação $x_2 \times x_4$ para o problema do projeto ótimo robusto de um robô industrial. . . . .	214
9.33	Relação $x_3 \times x_4$ para o problema do projeto ótimo robusto de um robô industrial. . . . .	214
9.34	Diagrama de Campbell genérico. . . . .	217
9.35	Modelo de elementos finitos para o rotor estudado . . . . .	217
9.36	Curvas de Pareto nominal e robustas obtidas pelo algoritmo MOFA para o problema de uma máquina rotativa. . . . .	219
9.37	Diagrama de Campbell para o valor médio . . . . .	221
9.38	Diagrama de Campbell para o caso nominal . . . . .	221
9.39	Diagrama de Campbell para o caso robusto . . . . .	222
9.40	Diagrama de Campbell para os três casos estudados. . . . .	222

# Lista de Tabelas

5.1	Principais estratégias para mutação e cruzamento no algoritmo DE . . . . .	92
5.2	Alguns dos Principais MOEA . . . . .	110
8.1	Parâmetros do algoritmo MOFA utilizados nas simulações. . . . .	149
8.2	Resultados das métricas CM e DM na simulação do problema teste 1 nominal.153	
8.3	Resultados das métricas CM e DM na simulação do problema teste 1 robusto.155	
8.4	Resultados das métricas CM e DM na simulação do problema teste 2 robusto.156	
8.5	Resultados das métricas CM e DM na simulação do problema teste 3 nominal.159	
8.6	Resultados das métricas CM e DM na simulação do problema teste 3 robusto.160	
8.7	Resultados das métricas CM e DM na simulação do problema teste 4 nominal.162	
8.8	Resultados das métricas CM e DM na simulação do problema teste 4 robusto.164	
8.9	Resultados das métricas CM e DM na simulação do problema teste 5 nominal.166	
8.10	Resultados das métricas CM e DM na simulação do problema teste 6 nominal.168	
9.1	Parâmetros do algoritmo MOFA utilizados nas simulações. . . . .	182
9.2	Pontos da curva de Pareto nominal para o projeto ótimo de duas barras. .	185
9.3	Pontos da curva de Pareto robustas para o projeto ótimo de duas barras para $\delta = 0,00075$ . . . . .	185
9.4	Pontos da curva de Pareto robustas para o projeto ótimo de duas barras para $\delta = 0,00125$ . . . . .	185
9.5	Pontos da curva de Pareto nominal para o projeto ótimo de uma viga em balanço. . . . .	189
9.6	Pontos da curva de Pareto robusta com $\delta = 0,0009$ para o projeto ótimo de uma viga em balanço. . . . .	190
9.7	Pontos da curva de Pareto robusta com $\delta = 0,0013$ para o projeto ótimo de uma viga em balanço. . . . .	190
9.8	Pontos da curva de Pareto nominal para o projeto ótimo de uma viga soldada.194	
9.9	Pontos da curva de Pareto robusta com $\delta = 0,1$ para o projeto ótimo de uma viga soldada. . . . .	194
9.10	Pontos da curva de Pareto robusta com $\delta = 0,15$ para o projeto ótimo de uma viga soldada. . . . .	194
9.11	Parâmetros do robô industrial usados nas simulações . . . . .	203

9.12	Planejamento de experimentos e respostas obtidas nas simulações para o projeto ótimo robusto de um robô industrial . . . . .	204
9.13	Coefficientes da aproximação polinomial e parâmetros estatísticos para o projeto ótimo de um robô industrial . . . . .	205
9.14	Pontos da curva de Pareto nominal para o projeto ótimo de um robo industrial. . . . .	210
9.15	Pontos da curva de Pareto robusta com $\delta = 0,1$ para o projeto ótimo de um robo industrial. . . . .	211
9.16	Pontos da curva de Pareto robusta com $\delta = 0,2$ para o projeto ótimo de um robo industrial. . . . .	211
9.17	Características geométricas dos discos da máquina rotativa. . . . .	218
9.18	Máxima distância entre as velocidades críticas para os casos robustos e nominal. . . . .	220

# Lista de Símbolos

## Glossário

ACV	-	Algoritmo de Colônia de Vagalumes.
AG	-	Algoritmos Genéticos.
BFGS	-	Broyden-Fletcher-Goldfarb-Shanno.
DFP	-	Davidon-Fletcher-Powell.
ED	-	Evolução Diferencial.
EP	-	Enxame de Partículas.
FEMEC	-	Faculdade de Engenharia Mecânica.
FLOPS	-	<i>Floating Point Operations Per Second.</i>
KKT	-	Karush-Kuhn-Tucker.
MODE	-	<i>Multiobjective Differential Evolution.</i>
MOEA	-	<i>Multiobjective Evolutionary Algorithm.</i>
MOFA	-	<i>Multiobjective Optimization Firefly Algorithm.</i>
MOGA	-	<i>Multiobjective Genetic Algorithm.</i>
MSD	-	Desvio Médio Quadrático.
NPGA	-	<i>Niched Pareto Genetic Algorithm.</i>
NSGA	-	<i>Nondominated Sorting in Genetic Algorithms.</i>
PMOGA	-	<i>Pareto Multiobjective Optimization Genetic Algorithm.</i>
POMO	-	Problema de Otimização Multiobjetivo.
POMOR	-	Problema de Otimização Multiobjetivo Robusto.
RS	-	Recozimento Simulado.
SPEA	-	<i>Strength Pareto Evolutionary Algorithm.</i>
UFU	-	Universidade Federal de Uberlândia.
VEGA	-	<i>Vector Evaluated Genetic Algorithms.</i>
ZDT	-	Zitzler-Deb-Thiele.

## Capítulo 2

$\mathbb{R}^n$	- Espaço vetorial real de dimensão $n$ .
$\vec{F}(x)$	- Vetor de objetivos.
$f_1(x), \dots, f_n(x)$	- Funções objetivo.
$h_i(x), g_j(x)$	- Restrições de igualdade e desigualdade respectivamente.
$x_1, \dots, x_n$	- variáveis de projeto.
$x_i^{inf}, x_i^{sup}$	- valores mínimo e máximo para a variável de projeto $x_i$ .
$\preceq$	- Relação de ordem genérica.
$x \triangleleft y$	- A solução $x$ domina a solução $y$ segundo Pareto.
$\mathcal{F}$	- Espaço de objetivos.
$\mathcal{H}$	- Espaço de projetos.
$P_{\text{ótimo}}$	- Conjunto ótimo de Pareto.
$P_{\text{front}}$	- Curva de Pareto.
$\partial(\mathcal{F})$	- Fronteira do espaço de objetivos.
$\mathcal{R}$	- Conjunto de soluções encontradas por um algoritmo genérico.
$T_{\text{erro}}$	- Métrica de convergência denominada Taxa de Erro.
$CM$	- Métrica de convergência denominada Convergência Métrica.
$\Gamma$	- Métrica de convergência denominada Distância Geracional.
$Spc$	- Métrica de diversidade denominada Espaçamento.
$NC$	- Métrica de diversidade denominada Número de Nichos.
$EM$	- Métrica de diversidade denominada Espalhamento Máximo.
$DM$	- Métrica de diversidade denominada Diversidade Métrica.
$d_f, d_l$	- Distância euclideana entre as soluções extremas dos conjuntos $P_{\text{front}}$ e $\mathcal{R}$ .

## Capítulo 3

$w_i$	- Pesos utilizados em ponderações.
$c_i$	- Constante normalizadora do objetivo $f_i$ .
$f_i^*$	- Valor ótimo da $i$ -ésima função objetivo.
$D_f$	- Domínio de definição da função $f$ .
$x_i^*$	- Ponto ótimo do objetivo $f_i$ .
$\xi$	- Valor da restrição para o método $\xi$ -restrito.

## Capítulo 4

$\mathbb{S}$	- Subconjunto do espaço $\mathbb{R}^n$ .
$\{x_k\}_{k \in \mathbb{N}}$	- Sequência de números reais.
$x_k$	- Iteração de um método sequencial.
$\varphi$	- Operador definido no espaço $\mathbb{R}^n$ .
$x_0$	- Semente inicial de um método.
$x^*$	- Solução de um problema de otimização.

$\epsilon$	- Erro admissível.
$\nabla f$	- Vetor gradiente da função $f$ .
$N$	- Número de pontos.
$d_k$	- Direção de descida na $k$ -ésima iteração.
$\alpha_k$	- Tamanho do passo na $k$ -ésima iteração.
$\mathcal{D}_f(x)$	- Conjunto das direções de descida da função $f$ no ponto $x$ .
$\langle \cdot, \cdot \rangle$	- Produto interno no espaço $\mathbb{R}^n$ .
$L_f(f(x_0))$	- Conjunto de nível.
$\mathcal{T}$	- Operador definido num espaço de Banach.
$\mathcal{B}$	- Espaço de Banach.
$f'(x)$	- Derivada ordinária de $f$ no ponto $x$ .
$J$	- Matriz jacobiana.
$\mathcal{H}_f$	- Matriz hessiana de $f$ .
$Q_k$	- Matriz simétrica positiva definida.
$r_p$	- Fator de penalidade.
$P(x)$	- Função de penalidade.
$\varphi(x, r_p)$	- Função pseudo-objetivo.

## Capítulo 5

$\rho$	- Número aleatório gerado no intervalo $[0, 1]$ .
$N_p$	- Tamanho da população.
$P_m$	- Probabilidade de mutação.
$F_p$	- Fator de perturbação.
$P_c$	- Probabilidade de cruzamento.
$x_\alpha, x_\beta, x_\gamma$	- Vetores no espaço de projetos.
$N_{var}$	- Número de variáveis.
$P_c$	- Probabilidade de cruzamento.
$x_{old}$	- Melhor vetor da geração anterior.
$x_{best}$	- Melhor vetor da geração corrente.
$\beta_0$	- Fator de atratividade máxima.
$\beta$	- Fator de atratividade.
$\gamma$	- Coeficiente de absorção da luz.
$p_i^{best}$	- Melhor posição da partícula $i$ .
$v_i$	- Velocidade atual da partícula $i$ .
$p_{best}^{swarm}$	- Melhor posição do enxame.



$\Delta E$	-	Varição de energia.
$T$	-	Temperatura.
$k_B$	-	Constante de Boltzman.
$Sh$	-	Função de compartilhamento.
$\sigma_{share}$	-	Raio de nicho.
$nc$	-	Contador de nicho.
$\mu(x_i)$	-	Número de soluções de mesmo <i>ranking</i> da solução $x_i$ .
$P$	-	População.
$d_i$	-	Distância de multidão da solução $x_i$ .
$F_i$	-	Pseudo-curva de Pareto.

## Capítulo 6

$\xi$	-	Fator de ruído.
$\Omega$	-	Subconjunto do espaço $\mathbb{R}^n$ .
$\alpha$	-	Vetor de parâmetros.
$\delta$	-	Fator de perturbação.
$\tilde{f}(x, \xi)$	-	Regularização robusta.
$B(x, \xi)$	-	Círculo de centro em $x$ e raio $\xi$ .
$f_{exp}$	-	Medida de expectância.
$p(\delta, \Delta\alpha)$	-	Função de densidade conjunta dos parâmetros de incerteza $\delta$ e $\Delta\alpha$ .
$f_{var}$	-	Medida de variância.
$f^{eff}$	-	Média Efetiva.

## Capítulo 7

$N_{obj}$	-	Número de objetivos.
$N_{ger}$	-	Número de gerações.
$N_{amt}$	-	Tamanho da amostra.
$N_{vag}$	-	Número de vagalumes da população.
$F_{pert}$	-	Fator de perturbação.
$\delta_i$	-	Coefficiente de robustez.
$N_{viol}$	-	Número de violações do espaço viável.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Fundamentos de Otimização Multiobjetivo</b>	<b>9</b>
2.1	A Formulação do Problema de Otimização Multiobjetivo . . . . .	11
2.2	Otimalidade segundo Edgeworth-Pareto . . . . .	13
2.2.1	Operador de Dominância de Pareto . . . . .	15
2.3	Metas para Otimização Multiobjetivo . . . . .	20
2.3.1	Métricas de Convergência e de Diversidade . . . . .	20
2.4	Metodologias para a Resolução de POMO . . . . .	25
2.4.1	Quanto ao tipo de Algoritmo Utilizado . . . . .	25
2.4.2	Quanto à forma de Tratamento . . . . .	26
<b>3</b>	<b>Tratamento de POMO - Métodos <i>A Priori</i></b>	<b>28</b>
3.1	Método da Soma Ponderada . . . . .	29
3.1.1	Interpretação Geométrica . . . . .	32
3.2	Método de Otimização Hierárquico . . . . .	33
3.3	Método $\xi$ -Restrito . . . . .	35
3.3.1	Conexão com o Método da Soma Ponderada . . . . .	37
3.4	Método Híbrido . . . . .	38
3.5	Método do Critério Global . . . . .	39
3.6	Método da Métrica Ponderada . . . . .	41
3.7	Método do Critério Ponderado Exponencial . . . . .	43
3.8	Método do Produto Ponderado ( <i>Weighted Product Method</i> ) . . . . .	43
<b>4</b>	<b>Métodos Clássicos de Otimização</b>	<b>45</b>
4.1	Noções de Convergência . . . . .	45
4.2	Taxas de Convergência . . . . .	48
4.2.1	Critérios de Parada . . . . .	50
4.3	Métodos de Otimização Unidimensional . . . . .	53
4.3.1	Método da Comparação de Pontos de Rede . . . . .	54
4.3.2	Método da Bissecção . . . . .	54

4.3.3	Método da Seção Áurea . . . . .	57
4.3.4	Método de Interpolação Polinomial . . . . .	59
4.4	Métodos de Descida . . . . .	60
4.5	Método de Newton e Métodos Quase-Newton . . . . .	68
4.5.1	O Método de Newton . . . . .	70
4.5.2	Métodos Quase-Newton . . . . .	73
4.6	Tratamento de Restrições . . . . .	74
<b>5</b>	<b>Métodos Heurísticos de Otimização</b>	<b>78</b>
5.1	Algoritmos Genéticos - AG . . . . .	79
5.1.1	Operador Seleção . . . . .	82
5.1.2	Operador Cruzamento . . . . .	84
5.1.3	Operador Mutação . . . . .	85
5.2	Evolução Diferencial . . . . .	86
5.2.1	Operador de Mutação . . . . .	88
5.2.2	Operador de Cruzamento . . . . .	89
5.2.3	Operador de Seleção . . . . .	90
5.2.4	Estratégias do Algoritmo DE . . . . .	91
5.3	Colônia de Vagalumes . . . . .	92
5.3.1	O Algoritmo de Colônia de Vagalumes . . . . .	93
5.4	Enxame de Partículas - EP . . . . .	95
5.4.1	O Algoritmo EP . . . . .	96
5.5	Recozimento Simulado - RS . . . . .	98
5.5.1	Descrição do Algoritmo RS . . . . .	99
5.6	Estudo de POMO via Métodos Heurísticos de Otimização . . . . .	101
5.6.1	Mecanismo de Avaliação da Aptidão . . . . .	102
5.6.2	Mecanismo de Corte de Soluções . . . . .	106
<b>6</b>	<b>Otimização Robusta</b>	<b>111</b>
6.1	Trabalhos Relacionados . . . . .	113
6.2	Representação dos Parâmetros de Incerteza . . . . .	116
6.3	Medidas de Robustez . . . . .	120
6.3.1	Regularização Robusta . . . . .	121
6.3.2	Medidas de Expectância e Variância . . . . .	122
6.3.3	Média Efetiva . . . . .	124
6.4	Otimização Robusta Multiobjetivo . . . . .	127
<b>7</b>	<b>Otimização Multiobjetivo via Colônia de Vagalumes</b>	<b>131</b>
7.1	Aspectos Gerais do Algoritmo . . . . .	131
7.2	O Algoritmo MOFA . . . . .	133
7.2.1	Geração da População Inicial . . . . .	134

---

7.2.2	Operador de Ordenamento . . . . .	135
7.2.3	Operador de Truncamento . . . . .	137
7.2.4	Operador de Seleção . . . . .	137
7.2.5	Operador de Perturbação . . . . .	138
7.2.6	Inserção do Parâmetro de Robustez - Aproximação da Média Efetiva	139
7.2.7	Tratamento das Restrições . . . . .	141
7.2.8	Critério de Parada . . . . .	143
7.3	Estruturação do MOFA . . . . .	144
<b>8</b>	<b>Aplicações Matemáticas</b>	<b>148</b>
8.1	Problema Teste 1 . . . . .	150
8.2	Problema Teste 2 . . . . .	155
8.3	Problema Teste 3 . . . . .	157
8.4	Problema Teste 4 . . . . .	160
8.5	Problema Teste 5 . . . . .	164
8.6	Problema Teste 6 . . . . .	167
8.7	Problema Teste 7 . . . . .	169
8.8	Análise de Sensibilidade dos Parâmetros do MOFA . . . . .	171
8.9	Conclusões Preliminares . . . . .	177
<b>9</b>	<b>Aplicações em Engenharia</b>	<b>181</b>
9.1	Problema de duas Barras . . . . .	182
9.2	Problema de uma Viga em Balanço . . . . .	187
9.3	Problema da Viga Soldada . . . . .	191
9.4	Projeto Ótimo Robusto de um Robô Industrial utilizando Metamodelos . .	198
9.4.1	Metamodelagem . . . . .	198
9.4.2	Projeto Ótimo Robusto de um Robô Industrial . . . . .	201
9.5	Problema em Dinâmica de Rotores . . . . .	215
9.5.1	Otimização do Eixo de um Rotor . . . . .	216
9.6	Conclusões Preliminares . . . . .	223
<b>10</b>	<b>Considerações Finais e Sugestões</b>	<b>224</b>
10.1	Conclusões . . . . .	224
10.1.1	Contribuições . . . . .	227
10.1.2	Sugestões para Trabalhos Futuros . . . . .	227
	<b>Referências Bibliográficas</b>	<b>229</b>

# Capítulo 1

## Introdução

A origem de problemas de otimização é muito remota e anterior ao desenvolvimento de ferramentas computacionais e matemáticas para a sua resolução. Pode-se citar, como exemplo, o problema de se encontrar qual a maior área que pode ser cercada por uma conhecida quantidade de corda (Problema da Princesa Dido). Na obra “Eneida de Virgílio” encontra-se uma referência a este problema: “No século IX antes de Cristo, a princesa fenícia Dido chegando às terras do norte da África junto com seu irmão Pigmalião, fizeram um acordo com os habitantes locais. Ao querer a princesa Dido comprar terra para se estabelecer com seu povo, o rei daquele lugar somente lhe permitiu comprar a parcela de terra que poderia ser cercada pela pele de um touro. Neste caso, a princesa Dido cortou a pele em pequenas tiras formando uma larga corda (entre 1000 a 2000 metros) e a dispôs de maneira que cobrisse a maior parte de terreno possível...” (BOYER;MERZBACH, 2011). A área que a princesa cercou tinha o formato de um círculo. Em 1870, o matemático K. Weirstrass apresentou uma solução para o problema baseando-se no Cálculo Variacional (MOREIRA; SALDANHA, 1993; FIGUEIREDO, 1989). Outro problema bastante interessante que também envolve otimização é o Problema da Braquistócrona, do grego

---

brakhisto (o mais curto) e chronos (tempo). O problema consiste em encontrar qual a trajetória que uma partícula deve percorrer com o menor tempo possível sendo conhecidos os pontos de saída e chegada, com velocidade inicial nula, sem atrito e sujeita apenas a ação da gravidade (BENNATON, 2001). Este problema foi proposto em 1696 pelo matemático suíço Johann Bernoulli, que deu um prazo de seis meses (estendido posteriormente para um ano) para que algum matemático apresentasse uma solução. Apenas cinco matemáticos apresentaram soluções, a saber: a do próprio Johann, uma de seu irmão Jakob, uma de Leibniz, outra de L'Hôpital e uma sem autor. Porém, Johann atribuiu a solução anônima à Isaac Newton, pois, segundo ele, “um leão é reconhecido pelas suas garras” (BOYER;MERZBACH, 2011).

À medida que foram sendo desenvolvidas as ferramentas matemáticas e com o aumento da popularidade das técnicas computacionais, aumentou também a complexidade dos problemas de otimização possíveis de serem considerados. Ao estudar um problema real em engenharia, é natural o engenheiro se deparar com muitas variáveis de projeto, várias restrições tecnológicas e/ou econômicas e, na maioria das vezes, com múltiplos objetivos. Deve ser salientado que a extensão do conceito de ótimo para o caso multiobjetivo não é trivial e, ao contrário do caso mono-objetivo, onde há apenas um ótimo global, em problemas multiobjetivo há uma curva solução, denominada **Curva de Pareto**, formada pelas soluções não dominadas ou não inferiores do problema. Além da convergência de um algoritmo, meta comum nos casos mono e multiobjetivo, uma meta específica para o caso multiobjetivo é a diversidade das soluções. Os pontos da Curva de Pareto devem estar bem espaçados, pois a diversidade garante uma gama de soluções distintas e que podem ser escolhidas, a depender do critério de pós-processamento utilizado.

O pioneiro na implementação de um algoritmo evolutivo multiobjetivo foi Schaffer (SCHAFFER, 1984). Após o trabalho de Schaffer, vários trabalhos de pesquisa têm

sido publicados com o intuito de fornecer novas técnicas de resolução de Problemas de Otimização Multiobjetivo (POMO) e também na aplicação das técnicas emergentes a problemas nas mais diversas áreas do conhecimento. Além disso, meios científicos tem difundido as novas metodologias e conceitos, bem como aplicações em diversas áreas de ciência. Vários pesquisadores tem se empenhado na publicação de trabalhos sobre otimização multicritério. Osyczka (1984) é uma referência clássica quando se trata de otimização multicritério. Em sua obra são tratados problemas multiobjetivo aplicados à engenharia, incluindo programas computacionais escritos na linguagem Fortran.

Sem dúvida a maioria dos algoritmos de otimização multiobjetivo, descritos e/ou utilizados na literatura corrente, são baseados nos algoritmos evolutivos. Porém, nos últimos anos, percebe-se o surgimento e difusão de meta-heurísticas que simulam o comportamento coletivo de algumas espécies, a saber, na genética de populações, nos processos físicos, químicos ou podem ser puramente estruturais (DEB, 2001; DOERNER et al., 2001; COELHO, 2004; BRANKE et al., 2008; LOBATO, 2008; LOBATO et al., 2011a).

Do ponto de vista prático, existem alguns problemas que dificultam a identificação de um ótimo para o problema de otimização, como por exemplo a multimodalidade. Uma questão interessante surge quando se deseja encontrar um ponto no espaço de projeto com uma alta precisão. Neste caso alguns aspectos podem ser ressaltados (LEIDEMER, 2009):

- mesmo que o ótimo verdadeiro seja localizado, talvez nunca possa ser possível implementá-lo na prática, pois existem incertezas associadas ao processo de construção. Além disso, até por se exigir um alto grau de precisão na fabricação, esta pode ser demasiadamente cara, portanto economicamente inviável;
- a formulação do problema de otimização é inerentemente estática. A realidade é

essencialmente dinâmica. Neste contexto, o problema pode estar associado à flutuação de parâmetros ambientais como temperatura, velocidade do vento, umidade ou mesmo pode haver desgaste de alguns componentes.

Diante do exposto, os sistemas a serem otimizados podem ser bastante sensíveis a pequenas alterações das variáveis de projeto, e assim, pequenas variações nas variáveis de projeto podem causar enormes variações dos objetivos (LEIDEMER, 2009), conforme argumentado anteriormente. Portanto, faz-se necessário encontrar uma metodologia que produza soluções que sejam pouco sensíveis a pequenas variações no projeto. Soluções com esta característica são chamadas de robustas e o procedimento para encontrar tais soluções é denominado **Otimização Robusta**. O apelo para a otimização robusta é que suas soluções e o desempenho dos resultados (funções objetivos) permaneçam relativamente sem mudanças quando expostas a certas condições de incerteza. A Otimização Robusta tem suas raízes na engenharia, sendo diretamente relacionado ao nome G. Taguchi, que é considerado o precursor desta área (TAGUCHI, 1984). Com o advento de computadores de alta velocidade e devido ao crescimento exponencial da taxa FLOPS (*Floating Point Operations Per Second*), a otimização de projetos robustos tem atraído crescente interesse nos últimos anos. Reflexo disto é uma edição da ASME *Journal of Mechanical Design*, inteiramente devotada a projetos robustos e também os trabalhos de Du, Wang e Chen (2000), Branke (2002), Park et al. (2006) e Beyer e Sendhoff (2007).

Segundo Paenk et. al (2006), de longe a maioria das atividades de pesquisa em otimização robusta se trata de problemas mono-objetivo. Também corroboram com esta afirmação alguns pesquisadores com atividade reconhecida na área de otimização evolucionária multiobjetivo. Deb e Gupta (2006) afirmam que, considerando o que conhecem sobre o tema, não há um estudo sistemático introduzindo robustez na otimização multiobjetivo. Sendo assim, é de fundamental importância o desenvolvimento e testes de



metodologias consistentes para o tratamento de problemas de otimização multiobjetivo robusta. Portanto, a presente tese, cujo principal objetivo é desenvolver uma metodologia consistente para resolver problemas de otimização multiobjetivo robusta, se apresenta como uma contribuição para esta área que ainda carece de estudos sistemáticos. Para a finalidade descrita, será empregado o Algoritmo de Colônia de Vagalumes (YANG, 2008) como ferramenta de otimização, associado ao conceito de Média Efetiva (DEB;GUPTA, 2006).

Os objetivos secundários do presente trabalho são:

- definir solução multiobjetivo robusta para diferentes tipos de problemas;
- entender a influência dos parâmetros de incerteza sobre o perfil da Curva de Pareto Robusta quando comparada à Curva de Pareto Nominal;
- testar a metodologia proposta em problemas matemáticos com solução conhecida;
- testar a metodologia proposta em problemas reais de engenharia.

Este trabalho surge naturalmente como consequência das atividades de pesquisa na área de Otimização de Projetos Mecânicos desenvolvidas na Faculdade de Engenharia Mecânica (FEMEC) da UFU. Os trabalhos que possuem maior afinidade com a atual proposta são apresentados a seguir em ordem cronológica. Saramago (1998) foi a pioneira entre os alunos da FEMEC da área de otimização e estudou a otimização de trajetórias de robôs manipuladores. Assis (1999) estudou técnicas mono e multiobjetivo aplicadas ao projeto e identificação de parâmetros de máquinas rotativas. Butkewitsch (1998) apresentou um estudo, por métodos clássicos, da otimização de componentes automotivos. Braga (1998), em sua dissertação, utilizou a heurística Algoritmos Genéticos para resolução de problemas mono-objetivo em engenharia. Oliveira (2005) realizou um estudo

teórico de técnicas, não baseadas no conceito de dominância, de resolução de problemas de otimização multiobjetivo. Santos (2007) estudou sobre o planejamento da trajetória de robôs através de elementos de dinâmica, controle e otimização. Lima (2007) usou o algoritmo NSGA II (*Nondominated Sorting in Genetic Algorithms* (SRINIVAS;DEB, 1994)) para a otimização multiobjetivo robusta de sistemas mecânicos na presença de amortecimento viscoelástico. Lobato (2008) apresentou na sua tese o desenvolvimento de um algoritmo multiobjetivo baseado na heurística Evolução Diferencial e realizou um estudo em aplicações matemáticas e nas engenharias mecânica e química. Viana (2008) desenvolveu, em ambiente Matlab<sup>®</sup>, o pacote *Simple Optimization Toolbox* com técnicas de metamodelagem aplicadas a problemas de otimização. Borges (2008) usou técnicas de otimização robusta no estudo sobre absorvedores dinâmicos de vibrações. Barros (2009) estudou um projeto de otimização de absorvedor dinâmico de vibrações multimodal. Oliveira (2012), utilizando técnicas determinísticas e heurísticas de otimização multiobjetivo, realizou um estudo sobre o projeto ótimo de robôs manipuladores 3r considerando a topologia do espaço de trabalho. Vicente (2014) estudou sobre um projeto robusto de circuitos *Shunt* para o controle passivo de vibrações de estruturas compostas.

A presente tese possui a seguinte estrutura:

- O Capítulo 2 apresenta os principais conceitos e definições relativas à otimização multiobjetivo, a saber, a definição de um POMO, o conceito de otimalidade de Pareto e as principais métricas de convergência e de diversidade;
- O Capítulo 3 apresenta as principais técnicas de tratamento de POMO que existem na literatura. As técnicas de tratamento de POMO apresentadas transformam o POMO em um problema mono-objetivo;
- O Capítulo 4 é dedicado a apresentação dos métodos clássicos (determinísticos) de

otimização. São apresentadas também as principais noções de convergência e taxas, assim como os critérios de parada. Os métodos descritos são uni e multidimensionais;

- No Capítulo 5 são descritas, de forma sucinta, as principais heurísticas usadas para resolver problemas de otimização. Associado ao conceito de não dominância, estas heurísticas são usadas para obter a curva de Pareto em uma única execução do algoritmo;
- O Capítulo 6 apresenta os principais conceitos e definições relativas à otimização robusta. Neste capítulo é definido o conceito de solução multiobjetivo robusta e são apresentadas as principais fontes de incertezas bem como as principais metodologias de incorporação da busca da robustez ao problema de otimização;
- O capítulo 7 apresenta o algoritmo *Multiobjective Optimization Firefly Algorithm* (MOFA). São descritas a atuação dos principais operadores do algoritmo bem como sua estrutura;
- No capítulo 8 a metodologia proposta é testada em problemas de otimização com solução analítica conhecida. Para cada problema é avaliada a convergência do algoritmo, a diversidade das soluções e o efeito do parâmetro de robustez no perfil da Curva de Pareto Robusta;
- No capítulo 9 o algoritmo MOFA é usado para resolver problemas de engenharia.
- Finalmente, no Capítulo 10, são apresentadas as conclusões e sugestões para trabalhos futuros.

## Capítulo 2

# Fundamentos de Otimização Multiobjetivo

Este capítulo é motivado pela descrição da teoria da otimização multiobjetivo. Não há a pretensão de esgotar o assunto e sim de explicar sobre as principais características e definições sobre a otimização multiobjetivo. Objetiva-se formular o problema de otimização multiobjetivo, definir soluções não dominadas, o operador de dominância de Pareto, bem como fazer uma descrição matemática das principais métricas de avaliação da qualidade (convergência e diversidade). Devido à crescente necessidade científica e tecnológica na busca de soluções que envolvem problemas de otimização com mais de um objetivo, o desenvolvimento e validação de metodologias para resolver problemas multiobjetivos, multicritérios ou de otimização vetorial tem atraído grande atenção do meio acadêmico(SCHAFFER, 1984; DEB, 2001; BABU et al., 2005; LOBATO, 2008).

Problemas de Otimização em Engenharia possuem dificuldades inerentes à área, pois vivenciam situações reais e cada vez mais complexas. São constituídos principalmente por sistemas de equações algébricas e diferenciais (parciais e ordinárias) que surgem a partir da caracterização física do problema. Em geral são problemas restritos, pois possuem restrições econômicas, ambientais, limitações físicas, etc. Outro ponto importante no

tratamento de Problemas de Otimização Multiobjetivo (POMO) é que, em quase sua totalidade, é impossível a obtenção de solução analítica de problemas reais.

Existem diferenças significativas entre problemas de otimização mono-objetivo e multiobjetivo. Segundo Deb (2001), as principais são:

- no problema mono-objetivo o ótimo é um ponto do espaço de projetos que minimiza (maximiza) a função objetivo. Em POMO o ótimo é uma curva formada, a priori, por infinitos pontos. É importante ressaltar que geralmente os objetivos do POMO são conflitantes, isto é, a melhora de um pode acarretar a piora de outros;
- na otimização mono-objetivo trabalha-se apenas com o espaço de projetos. No caso multiobjetivo trabalha-se simultaneamente com dois espaços; o de projetos e o de objetivos. Manter a diversidade de soluções no espaço de objetivos é fundamental para a qualidade de soluções, pois ajuda na tomada de decisões para a escolha do ponto a ser implementado na prática (pós-processamento). Outro detalhe importante é que nem sempre é possível garantir que a proximidade de soluções no espaço de projetos implique na proximidade de soluções no espaço de objetivos.

Os principais algoritmos para a obtenção da solução de um POMO utilizam recursos determinísticos e heurísticos. O tratamento determinístico é baseado no cálculo diferencial e algumas das principais técnicas serão descritas no capítulo 4 deste trabalho. Os heurísticos são métodos que simulam o comportamento coletivo de algumas espécies, a saber, na genética de populações, nos processos físicos, químicos ou podem ser puramente estruturais (BRANKE et al., 2008). Algumas estratégias heurísticas serão descritas no capítulo 5. Os métodos determinísticos são métodos de busca ponto-a-ponto e por isto não é possível obter a solução de um POMO em apenas uma única execução. Além disso, aplicações sucessivas de métodos determinísticos não garantem uma boa aproximação da

solução e nem tampouco a diversidade das soluções (LOBATO, 2008). Assim os métodos heurísticos que trabalham com uma população de pontos consistem em metodologias mais apropriadas na resolução de problemas e otimização multiobjetivo, pois podem alcançar a solução do POMO em uma única execução (DEB, 2001).

O pioneiro na implementação de um algoritmo evolutivo multiobjetivo foi Schaffer que em 1984, em sua tese de doutorado, utilizou a heurística Algoritmos Genéticos em POMO (SCHAFFER, 1984). Gandibleux et al. (1997) utilizam a heurística Busca Tabu para resolver problemas multiobjetivo combinatoriais. Fang et al. (2005) realizam um estudo comparativo de métodos de metamodelagem para problemas multiobjetivo de resistência à colisão. Doerner et al. (2001) utilizam a metaheurística Otimização por Colônia de Formigas para um problema multiobjetivo de seleção de portfólio. Coelho (2004) utiliza a otimização multiobjetivo na otimização de projetos mecânicos. Deb (2001) realiza, em seu livro, um estudo sobre otimização multiobjetivo utilizando conceitos de otimização evolutiva. Lobato (2008) utilizou um algoritmo multiobjetivo envolvendo a heurística Evolução Diferencial para a solução de problemas de projeto de sistemas em engenharia.

## 2.1 A Formulação do Problema de Otimização Multiobjetivo

De acordo com Vanderplaats (1999) um problema de otimização é constituído pelos seguintes elementos:

- **Função Objetivo** - define a característica do sistema a ser otimizado. Esta é a função que relaciona a variável dependente com as variáveis independentes (variáveis de projeto);
- **Variáveis de Projeto** - são as variáveis independentes que influenciam a função objetivo e as restrições. Variações dos valores destas variáveis provocam o aumento ou

diminuição nos valores da função objetivo. Também são chamadas de variáveis de busca ou de decisão;

- **Restrições** - são as limitações do projeto a ser otimizado. Estas podem ocorrer devido a limitações físicas, de construção (econômicas), ambientais, entre outras. As restrições determinam uma região chamada de Espaço de Projetos, que é um subespaço do espaço  $n$ -dimensional  $\mathbb{R}^n$ , onde  $n$  é o número de variáveis de projeto. As restrições podem ser de três tipos: restrições de igualdade, restrições de desigualdade e restrições laterais.

Matematicamente o POMO pode ser definido como (DEB, 2001):

$$\begin{aligned} \text{minimizar} \quad & \vec{F}(x) = (f_1(x), f_2(x), \dots, f_m(x)), \quad x \in \mathbb{R}^n & (2.1) \\ \text{sujeito a} \quad & \begin{cases} h_l(x) = 0, & l = 1, \dots, L \\ g_j(x) \leq 0, & j = 1, \dots, J \\ x_i^{inf} \leq x_i \leq x_i^{sup}, & i = 1, \dots, n. \end{cases} & (2.2) \end{aligned}$$

onde:

- i*) as funções  $f_1, \dots, f_m$  são as  $m$  funções objetivo do problema;
- ii*)  $h_l(x)$  é o vetor das restrições de igualdade;
- iii*)  $g_j(x)$  é o vetor das restrições de desigualdade;
- iv*)  $x_i^{inf}$  e  $x_i^{sup}$  é o vetor das restrições laterais que determinam os limites inferior e superior, respectivamente, de variação da coordenada  $x_i$  do vetor  $x$ , para  $i = 1, \dots, n$ .

Em um problema real de engenharia, cada uma das funções  $f_i(x)$  ( $i = 1, \dots, m$ ) poderiam ser minimizadas ou maximizadas. Porém não há diferença, do ponto de vista

---

matemático, em um problema de minimização ou de maximização. Um pode ser transformado no outro de forma simples. Assim, ao longo desta tese um problema de otimização será tratado, como é feito na literatura corrente, apenas como um problema de minimização.

## 2.2 Otimalidade segundo Edgeworth-Pareto

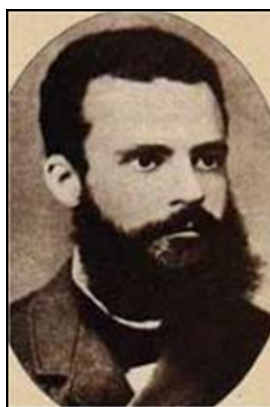
O conceito de ótimo para problemas com mais de um objetivo foi inicialmente concebido pelo economista inglês Francis Ysidro Edgeworth (EDGEWORTH, 1881, Fig. 2.1 - a) e posteriormente aperfeiçoado pelo também economista, mas agora italiano, Vilfredo Pareto (PARETO, 1896, Fig. 2.1 - b). Para entender o conceito tem-se que definir previamente o conceito de dominância de uma solução sobre outra solução. A maior parte dos algoritmos de otimização multiobjetivo utilizados pelos pesquisadores utilizam o conceito de dominância na busca pelas soluções ótimas (OSYCZKA, 1984). Por isto o entendimento do conceito de ótimo de Pareto e da definição de dominância é fundamental para compreender os algoritmos de otimização multiobjetivo.

Ao contrário do que é feito para o caso mono-objetivo, onde é possível definir matematicamente a solução ótima, no caso multiobjetivo não há uma definição de solução ótima e sim um conceito (entendimento) de solução ótima. Este conceito é conhecido por Postulado de Edgeworth-Pareto ou mais popularmente, na literatura, por Postulado de Pareto. A noção intuitiva do postulado de Pareto é que um ponto para ser solução de um problema de otimização multiobjetivo deve satisfazer a condição de que “*nenhum objetivo adotado melhora a solução sem piorar pelo menos um outro objetivo*” (LOBATO, 2008). Para encontrar pontos pertencentes à curva solução multiobjetivo é necessário entender quando um ponto, com mais de uma coordenada, é melhor que outro ponto. Este





(a) Francis Ysidro Edgeworth.



(b) Vilfredo Pareto.

Figura 2.1: Formuladores do conceito de ótimo multiobjetivo. Reproduzido de [http : //www.business.baylor.edu/SteveGardner/Neoclassicals.htm](http://www.business.baylor.edu/SteveGardner/Neoclassicals.htm) (acesso em 22 de janeiro de 2015).

entendimento é obtido através do conceito de dominância. Para entender quando é que uma solução domina outra solução considere um problema hipotético de minimização de dois objetivos  $f_1$  e  $f_2$ . A Figura 2.2 abaixo ilustra as possíveis escolhas entre seis soluções no espaço de objetivos.

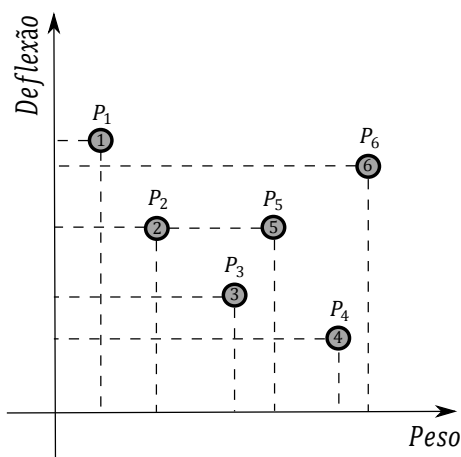


Figura 2.2: Opções de escolha de soluções em uma situação hipotética.

A situação hipotética considerada é de um problema de minimização dos objetivos  $f_1$  e  $f_2$ , sendo assim pode-se descartar o ponto  $P_6$ , pois claramente percebe-se, por exemplo,

que  $P_2, P_3, P_4$  e  $P_5$  são melhores. Na linguagem corrente da literatura de otimização diz-se que  $P_6$  é dominado por  $P_2, P_3, P_4$  e  $P_5$ . Apesar de  $P_5$  ser melhor do que  $P_6$ , os pontos  $P_2$  e  $P_3$  são preferíveis em relação a  $P_5$ , isto é,  $P_5$  é dominado por  $P_2$  e  $P_3$ . Por outro lado,  $P_3$  é melhor que  $P_5$  em ambos objetivos e  $P_2$ , apesar do objetivo  $f_2$  ser igual ao do ponto  $P_5$ , o objetivo  $f_1$  é menor. Pode-se observar que entre os pontos  $P_1, P_2, P_3$  e  $P_4$  não é possível afirmar, caso não haja nenhuma prioridade entre o atendimento de algum dos objetivos, qual solução é melhor do que a outra. O que é observado é que um ponto supera o outro em algum objetivo, porém perde em qualidade no outro objetivo. Como não há soluções melhores que  $P_1, P_2, P_3$  e  $P_4$ , diz-se então que estes são não dominados, não inferiores, eficientes ou soluções Ótimas de Pareto (BRANKE et al., 2008).

Segundo Deb (2001), o espaço de objetivos, em um problema de otimização, é decomposto em dois subconjuntos disjuntos: o conjunto de pontos dominados e o conjunto dos não dominados. Estes conjuntos são caracterizados pelas propriedades:

- i)* qualquer elemento do conjunto de pontos dominados deve ser dominado por pelo menos um elemento do conjunto de pontos não dominados;
- ii)* nenhum elemento do conjunto de pontos não dominados é dominado por outro elemento do conjunto de pontos não dominados.

## 2.2.1 Operador de Dominância de Pareto

Esta seção destina-se à definição de um operador de comparação entre dois pontos no espaço de objetivos. Esta comparação é usada para determinar quando um ponto deve ser preferido em relação a outro ponto. Muitos algoritmos computacionais de otimização multiobjetivo (evolutivos, estruturais e baseados em colônias) utilizam o conceito de dominância de Pareto na busca da solução para o POMO.

Para haver convergência, um algoritmo multiobjetivo deve escolher os melhores pontos do espaço de objetivos. Assim, faz-se necessário estabelecer uma relação de ordem neste espaço, que necessariamente possui dimensão maior que um. Seguem abaixo as definições necessárias para o entendimento do critério de preferência de um ponto sobre outro.

**Definição 2.1.** *Uma relação binária  $\preceq$  sobre um conjunto  $\mathcal{M}$  é dita relação de ordem parcial se forem verdadeiras as seguintes propriedades:*

- |  |                                   |
|--|-----------------------------------|
| a) $x \preceq x$ para todo $x \in \mathcal{M}$   | <b>Propriedade Reflexiva</b>      |
| b) se $x \preceq y$ e $y \preceq x$ então $x = y$ , onde $x, y \in \mathcal{M}$          | <b>Propriedade Anti-Simétrica</b> |
| c) se $x \preceq y$ e $y \preceq z$ então $x \preceq z$ , onde $x, y, z \in \mathcal{M}$ | <b>Propriedade Transitiva</b>     |

**Definição 2.2.** *Quando há uma relação de ordem parcial  $\preceq$  sobre um conjunto  $\mathcal{M}$  o conjunto em questão é dito parcialmente ordenado. Dois elementos  $x, y \in \mathcal{M}$  são ditos comparáveis sempre que  $x \preceq y$  ou  $y \preceq x$ . Se todos pontos do conjunto  $\mathcal{M}$  forem comparáveis segundo a relação  $\preceq$ , então a relação é dita total.*

Em um problema multiobjetivo onde deseja-se minimizar os objetivos é importante que esteja bem claro quando um vetor de objetivos, aplicado em um ponto, é *menor ou igual* que o mesmo vetor de objetivos aplicado em outro ponto. Assim é necessário definir uma relação de ordem nos conjuntos  $\mathbb{R}^n$ , para  $n \geq 2$ , isto é, estender o conceito de menor ou igual para espaços  $\mathbb{R}^n$  com  $n \geq 2$ . Define-se uma relação de ordem no espaço  $\mathbb{R}^n$  da seguinte forma:

**Teorema 2.3.** *Sejam  $x = (x_1, x_2, \dots, x_n), y = (y_1, y_2, \dots, y_n) \in \mathbb{R}^n$ . A relação binária  $\preceq$  em  $\mathbb{R}^n$  definida por*

$$x \preceq y \implies x_i \leq y_i, \quad i = 1, 2, \dots, n \quad (2.3)$$

onde o símbolo  $\leq$  representa a relação menor ou igual usual dos números reais é uma relação de ordem parcial em  $\mathbb{R}^n$ .

*Demonstração.* Ver em Domingues e Iezzi (2003). □

A relação  $\preceq$  é uma relação de ordem parcial em  $\mathbb{R}^n$ . Silva (2003) demonstrou em seu artigo que não é possível definir uma relação de ordem total sobre conjuntos  $\mathbb{R}^2$ , porém não é uma tarefa difícil estender este resultado para espaços  $\mathbb{R}^n$  com  $n > 2$ . Assim qualquer relação de ordem definida sobre  $\mathbb{R}^n$  com  $n \geq 2$  será apenas parcial. A seguir será definido o operador de dominância de Pareto que é bastante utilizado em algoritmos de otimização multiobjetivo como critério de preferência de uma solução sobre outra (DEB, 2001).

**Definição 2.4.** *Sejam  $x, y \in \mathbb{R}^n$ . Diz-se que a solução  $x$  domina a solução  $y$ , no espaço de projetos, e escreve-se  $x \triangleleft y$ , se as condições forem satisfeitas (no espaço de objetivos):*

i)  $\vec{F}(x) \preceq \vec{F}(y)$ , isto é, se  $f_i(x) \leq f_i(y)$  para  $i = 1, 2, \dots, m$ ;

ii)  $\vec{F}(x) \neq \vec{F}(y)$ , isto é, existe pelo menos um  $i$  tal que  $f_i(x) < f_i(y)$ .

A condição i) diz que, para nenhum dos objetivos, a solução  $x$  é maior (no sentido ordinário) do que a solução  $y$ . A condição ii) diz que pelo menos em um dos objetivos da solução  $x$  é estritamente menor do que o da solução  $y$ . Assim, se as duas condições acima forem satisfeitas, diz-se que a solução  $y$  é dominada pela solução  $x$ , ou que a solução  $x$  é não dominada pela solução  $y$ . Em um problema de otimização multiobjetivo as soluções não dominadas são preferíveis em relação às dominadas. A seguir é apresentada a definição de dominância forte de uma solução sobre outra.

**Definição 2.5.** *Uma solução  $x$  domina fortemente outra solução  $y$ , no espaço de projetos, se  $\vec{F}(x) \prec \vec{F}(y)$ , isto é, se  $f_i(x) < f_i(y)$  para todo  $i = 1, 2, \dots, m$ .*

A seguir serão apresentadas as definições de conjunto ótimo de Pareto e também de Fronteira de Pareto. Considere os seguintes conjuntos: o espaço de projetos  $\mathcal{H}$  de um problema de otimização multiobjetivo e o espaço de objetivos  $\mathcal{F} = \vec{F}(\mathcal{H})$  que é a imagem, pelo vetor de objetivos  $\vec{F}$ , do espaço de projetos  $\mathcal{H}$ .

**Definição 2.6.** *O Conjunto Ótimo de Pareto, denominado por  $P_{\acute{o}timo}$ , é o subconjunto de  $\mathcal{H}$  formado por todas soluções não dominadas do espaço de projetos, quando comparadas a todas as soluções de  $\mathcal{H}$ , isto é,*

$$P_{\acute{o}timo} = \{x \in \mathcal{H}; x \triangleleft y, \forall y \in \mathcal{H}\}$$

**Definição 2.7.** *O Conjunto Curva de Pareto, denominado por  $P_{front}$ , é o subconjunto de  $\mathcal{F}$  formado por todas soluções não dominadas do espaço de objetivos, quando comparadas a todas as soluções de  $\mathcal{F}$ , isto é,  $P_{front} = \vec{F}(P_{\acute{o}timo})$*

Pode ser provado que a Curva de Pareto  $P_{front}$  está, no sentido matemático de topologia, na fronteira do conjunto  $\mathcal{F}$ , ou seja, realmente pode-se afirmar que  $P_{front} \subset \partial(\mathcal{F})$ , onde o conjunto  $\partial(\mathcal{F})$  é a fronteira do espaço de objetivos  $\mathcal{F}$ . A Figura 2.3 a seguir ilustra casos de Curvas de Pareto para problemas bi-dimensionais.

**Definição 2.8.** *Seja  $P_1 \subset \partial(\mathcal{F})$  um conjunto com a seguinte propriedade: Existe  $P_2 \subset \mathcal{F}$  tal que  $P_1 \subset P_2$  e qualquer solução de  $P_1$  domina todas soluções de  $P_2$ . Um conjunto  $P_1$  com esta propriedade é chamado de Curva de Pareto Local.*

A Figura 2.4, traz a interpretação geométrica para um caso onde o espaço de objetivos possui Curva de Pareto Local.

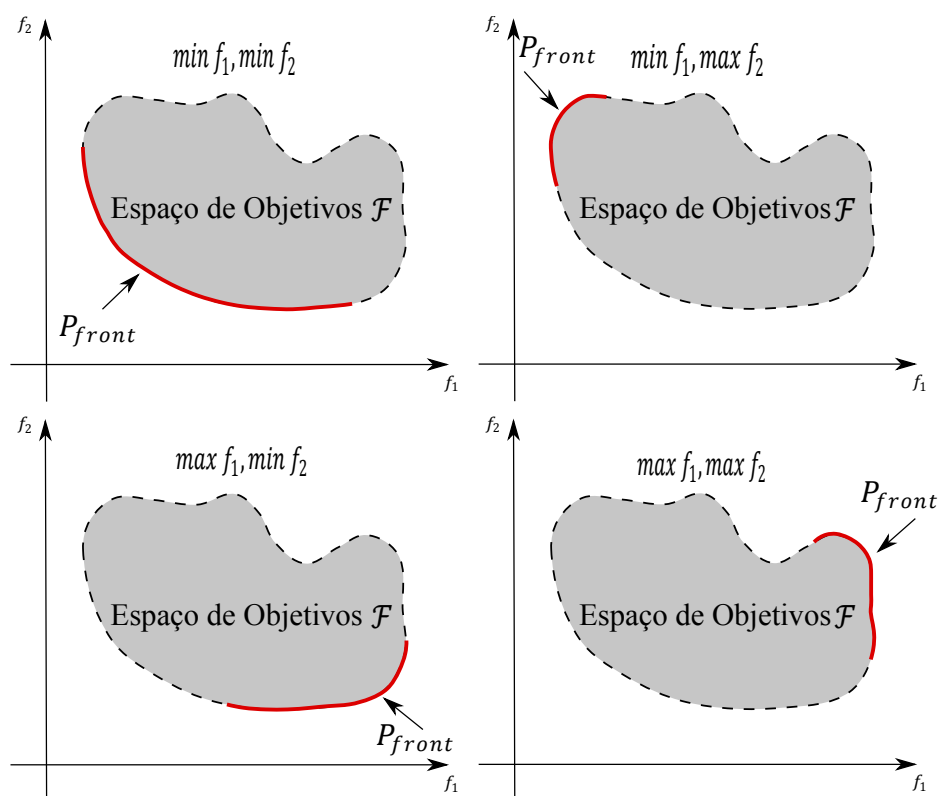


Figura 2.3: Curvas de Pareto para problemas com dois objetivos. Adaptado de (DEB, 2001).

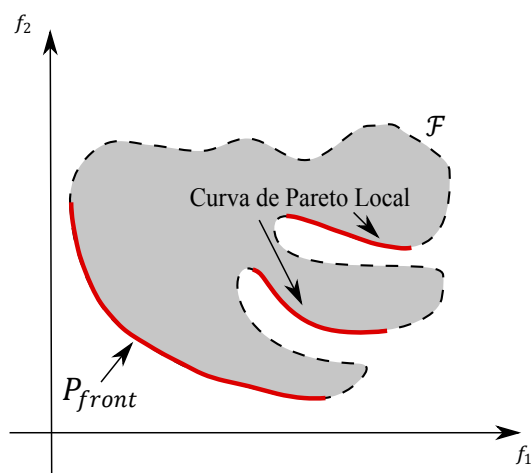


Figura 2.4: Geometria do espaço de objetivos com curva de Pareto local. Adaptado de (DEB, 2001).

## 2.3 Metas para Otimização Multiobjetivo

Segundo Deb (2001) a qualidade de um algoritmo para resolução de um POMO é medida basicamente através do alcance da qualidade das seguintes metas: convergência e diversidade. Segundo Lobato (2008) assegurar uma maior cobertura sobre a fronteira de Pareto indica que o conjunto das soluções está *comprometido* com os objetivos. Pontos bem próximos, no espaço de objetivos, a princípio podem ser considerados iguais do ponto de vista de escolha para a execução de um projeto real de engenharia. Assim, faz-se necessário o controle da diversidade de soluções em um algoritmo de otimização multiobjetivo. Outro fator, que é logicamente importante, é a convergência do algoritmo. A Figura 2.5, ilustra a atuação das metas mencionadas anteriormente.

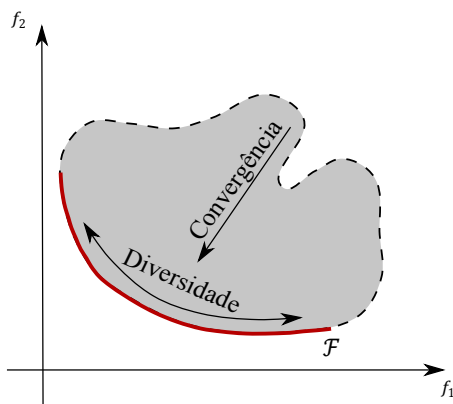


Figura 2.5: Metas em otimização multiobjetivo: convergência e diversidade. Adaptada de (DEB, 2001).

### 2.3.1 Métricas de Convergência e de Diversidade

Como mencionado na seção anterior, a qualidade da solução obtida por um algoritmo multiobjetivo é medida tanto pela sua convergência, em relação à solução analítica, quanto pela distribuição das soluções sobre a Curva de Pareto. As principais métricas usadas

dividem-se em: *Métricas de Convergência* e *Métricas de Diversidade*.

### Métricas de Convergência

As métricas de convergência caracterizam-se por fornecerem uma medida para a distância entre o conjunto de soluções  $\mathcal{R}$ , obtidas por um algoritmo, e o conjunto de soluções analíticas  $P_{front}$  (Curva de Pareto) de um problema de otimização multiobjetivo. A seguir serão apresentadas algumas métricas definidas por Deb (2001):

- a) Taxa de Erro ( $T_{erro}$ ) - Esta métrica conta quantas soluções do conjunto de soluções obtidas  $\mathcal{R}$  pertencem à Fronteira de Pareto  $P_{front}$ . Sua descrição matemática pode ser dada por:

$$T_{erro} = \frac{\sum_{i=1}^{|\mathcal{R}|} e_i}{|\mathcal{R}|} \quad \text{onde } e_i = \begin{cases} 1, & \text{se } x_i \notin P_{front} \\ 0, & \text{se } x_i \in P_{front} \end{cases} \quad (2.4)$$

O símbolo  $|\mathcal{R}|$  significa a quantidade de elementos no conjunto  $\mathcal{R}$ . Tem-se que  $0 \leq T_{erro} \leq 1$ . Quanto menor for o valor desta métrica, melhor será a convergência do algoritmo e quando  $T_{erro} = 0$  tem-se que  $\mathcal{R} \subset \mathcal{F}$ . A principal desvantagem desta métrica é que ela não fornece informações sobre a localização das soluções de  $\mathcal{R}$  que estão fora da Curva de Pareto, ou seja, algumas soluções podem estar bem distantes.

- b) Convergência Métrica ( $CM$ ) - Esta métrica calcula a média aritmética entre as distâncias das soluções  $\mathcal{R}$ , não dominadas, obtidas pelo algoritmo e o conjunto analítico  $P_{front}$ . Sua expressão matemática é dada por:

$$CM = \frac{\sum_{i=1}^{|\mathcal{R}|} d_i}{|\mathcal{R}|} \quad (2.5)$$



Onde  $d_i$  é a distância euclideana (no espaço de objetivos) entre a solução  $x_i \in \mathcal{R}$  e a solução mais próxima em  $P_{front}$ , isto é,

$$d_i = \underbrace{\min}_{x_k \in \mathcal{F}} \sum_{m=1}^n \sqrt{(f_m(x_k) - f_m(x_i))^2} \quad (2.6)$$

onde  $n$  é o número de objetivos do problema.

Um ponto interessante desta métrica é o fato dela considerar todas soluções  $\mathcal{R}$ , obtidas pelo algoritmo, no cálculo da média, ao contrário da métrica anterior. Observa-se que quanto menor for o valor da métrica  $CM$ , melhor é a convergência do algoritmo.

- c) Distância Geracional ( $\Gamma$ ) - Esta métrica, como a anterior, também calcula uma média aritmética das distâncias entre as soluções obtidas  $\mathcal{R}$  e a solução analítica  $P_{front}$ . Sua expressão matemática é dada por:

$$\Gamma = \frac{\left( \sum_{i=1}^{|\mathcal{R}|} d_i^p \right)^{1/p}}{|\mathcal{R}|} \quad (2.7)$$

onde  $p$  é um número natural não nulo. Quando  $p = 1$  tem-se que  $\Gamma = CM$ . Outro ponto interessante a se observar é quando  $p = 2$ , tem-se que

$$\Gamma = \frac{\left( \sum_{i=1}^{|\mathcal{R}|} d_i^2 \right)^{1/2}}{|\mathcal{R}|} = \frac{\sqrt{d_1^2 + d_2^2 + \dots + d_{|\mathcal{R}|}^2}}{|\mathcal{R}|} \leq \frac{\sqrt{d_1^2} + \sqrt{d_2^2} + \dots + \sqrt{d_{|\mathcal{R}|}^2}}{|\mathcal{R}|} = CM$$

### Métricas de Diversidade

Quando não é estabelecida uma prioridade de atendimento entre os objetivos, todos pontos da Curva de Pareto são igualmente ótimos. Assim, para que o projetista possa fazer

uma escolha posterior de uma solução particular, é de fundamental importância que as soluções possam estar bem distribuídas sobre a Curva de Pareto, isto é, que haja diversidade no conjunto solução. A diversidade no conjunto solução é uma meta específica de problemas de otimização multiobjetivo (Deb, 2001). A seguir serão apresentadas algumas das principais métricas de diversidade encontradas na literatura.

- a) Espaçamento ( $Spc$ ) - Proposta por Zitzler et al. (2000), esta métrica mede o quanto as soluções do conjunto  $\mathcal{R}$  estão espalhadas. Sua expressão matemática é dada por

$$Spc = \frac{\left(\sum_{i=1}^{|\mathcal{R}|} (d_i - \bar{d})^2\right)^{\frac{1}{2}}}{|\mathcal{R}|} \quad (2.8)$$

onde  $\bar{d}$  é a média aritmética entre as distâncias  $d_i$ , sendo que  $d_i$  é uma distância, no espaço de objetivos, entre a solução  $x_i \in \mathcal{R}$  e a solução em  $\mathcal{R}$  que esteja mais próxima de  $x_i$ , dada por:

$$d_i = \underbrace{\min}_{x_k \in \mathcal{R}, x_k \neq x_i} \sum_{m=1}^n |f_m(x_k) - f_m(x_i)| \quad (2.9)$$

sendo  $n$  o número de objetivos do problema.

Observa-se que esta métrica calcula o desvio padrão entre as distâncias de duas soluções consecutivas. Assim, quanto menor for o valor deste desvio, melhor distribuídas estarão as soluções do conjunto  $\mathcal{R}$ .

- b) Número de Nichos ( $NC$ ) - Esta métrica calcula o número de nichos existentes no conjunto de soluções  $\mathcal{R}$  obtidas pelo algoritmo. Sua expressão matemática é dada

por (Zitzler et al., 2000):

$$NC = \frac{1}{|\mathcal{R}| - 1} \sum_{i=1}^{|\mathcal{R}|} |x_j \in \mathcal{R}; d(x_i, x_j) > \sigma| \quad (2.10)$$

onde  $d(x_i, x_j)$  é a distância entre as soluções  $x_i$  e  $x_j$  que pode ser calculada no espaço de objetivos ou no de projetos. Quando  $d(x_i, x_j) < \sigma$  tem-se que as soluções  $x_i$  e  $x_j$  estão no mesmo nicho. Quanto maior o número de nichos formados melhor será a diversidade entre as soluções. Um ponto negativo desta métrica está na definição do parâmetro  $\sigma$ . Valores muito altos do parâmetro  $\sigma$  implicam em um número reduzido da quantidade de nichos, enquanto a situação inversa é observada para valores muito pequenos de  $\sigma$ .

- c) Espalhamento Máximo ( $EM$ ) - Esta é mais uma métrica proposta por Zitzler et al. (2000). Esta métrica retorna a distribuição das soluções no conjunto de soluções  $\mathcal{R}$  obtidas pelo algoritmo. Sua expressão é:

$$EM = \sqrt{\sum_{k=1}^n \left( \underbrace{\max_{x_j \in \mathcal{R}} f_k(x_j)} - \underbrace{\min_{x_j \in \mathcal{R}} f_k(x_j)} \right)}. \quad (2.11)$$

Quanto maior for o valor de  $EM$  significa que há uma melhor distribuição das soluções sobre o conjunto  $\mathcal{R}$ .

- d) Diversidade Métrica ( $DM$ ) - Proposta por Deb (2001), esta métrica também é utilizada para medir o espalhamento das soluções do conjunto  $\mathcal{R}$ . Sua expressão é dada por:

$$DM = \frac{d_f + d_l + \sum_{i=1}^{|\mathcal{R}|} |d_i - \bar{d}|}{d_f + d_l + (|\mathcal{R}| - 1)\bar{d}} \quad (2.12)$$

---

onde  $d_f$  e  $d_l$  representam a distância euclidiana entre as soluções extremas do conjunto ótimo de Pareto  $P_{front}$  e do conjunto de soluções não dominadas  $\mathcal{R}$ . O parâmetro  $d_i$  é a distância euclidiana, no espaço de objetivos, entre a solução  $x_i \in \mathcal{R}$  e o ponto mais próximo da fronteira  $P_{front}$  e  $\bar{d}$  é a média aritmética das distâncias  $d_i$ . Quanto menor for o valor da métrica DM há uma maior cobertura do conjunto  $\mathcal{R}$ .

## 2.4 Metodologias para a Resolução de POMO

Segundo Horn (1997), as principais metodologias para resolução de POMO dependem do tipo de algoritmo utilizado para encontrar a curva solução e também da forma adotada para o tratamento das soluções. Nas próximas seções deste capítulo serão apresentados os princípios que regem cada uma destas abordagens.

### 2.4.1 Quanto ao tipo de Algoritmo Utilizado

Segundo Saramago (1999), quanto ao tipo de algoritmo utilizado para a resolução de um POMO, as metodologias classificam-se em:

- **Clássica ou Determinística** - as técnicas pertencentes a esta classe são baseadas no cálculo diferencial. Com o desenvolvimento de técnicas do Cálculo Variacional e com o aumento da popularidade dos recursos computacionais, os métodos determinísticos passaram a ser bastante utilizados na literatura, inclusive em *softwares* comerciais. Seus pontos positivos estão na rápida convergência e garantia matemática de convergência para o ótimo quando a semente inicial encontra-se em uma região ótima. Seus pontos negativos estão na incapacidade de lidar com descontinuidades, com a

---

multimodalidade dos objetivos, com problemas discretos e com funções não convexas;

- **Aleatória ou não Determinística** - as técnicas pertencentes a esta classe são baseadas no comportamento coletivo de algumas espécies, na genética de populações, em analogias a processos físicos e químicos ou são algoritmos puramente estruturais. Na literatura as técnicas aleatórias mais famosas são os chamados Algoritmos Genéticos (GOLDBERG, 1989). Segundo Branke et al. (2008) as técnicas evolutivas de otimização tem se tornado, nos últimos anos, populares e de grande utilidade nas pesquisas e aplicações. São métodos baseados em população e que não utilizam nenhuma informação sobre derivadas do vetor de objetivos na sua implementação. São capazes de lidar com multimodalidade, problemas com variáveis contínuas, discretas e mistas, funções não convexas e descontinuidade dos objetivos. Na sua grande maioria trabalham com populações, sendo assim exploram melhor o espaço de projetos na busca por soluções ótimas, quando comparados com métodos ponto-a-ponto.

## 2.4.2 Quanto à forma de Tratamento

Segundo Coelho (2004) e Zitzler (1999) os métodos para tratamento de POMO classificam-se em três categorias:

- métodos *A Priori* - são métodos em que o projetista, antes do início processo de otimização, atribui elementos de preferência aos objetivos ou os classificar ordinalmente (hierarquizá-los). No primeiro passo o problema multiobjetivo é transformado em um problema mono-objetivo via agregação dos objetivos. No segundo passo resolve-se o problema considerando apenas o primeiro objetivo na ordem de hierarquia. A seguir, o problema é resolvido para o segundo objetivo, porém com uma restrição

associada à solução encontrada para o primeiro objetivo. Repete-se este processo até que se esgotem todos os objetivos. São as abordagens mais simples e óbvias entre as abordagens para tratamento de POMO (LOBATO, 2008).

- métodos *Progressivos* - são métodos utilizados durante o processo de busca de soluções ótimas. Nestas abordagens a interferência ocorre para subordinar as preferências para que as soluções *visitem* regiões mais promissoras do espaço de busca (soluções viáveis). Não necessariamente utilizam a agregação dos objetivos em um único objetivo. Os métodos progressivos podem utilizar o conceito de dominância de Pareto na busca por soluções ótimas. Segundo Branke et al. (2008) e Coelho (2004) são abordagens de fácil implementação computacional e há muitos códigos disponíveis para download nas páginas de vários institutos de pesquisa e pessoais de alguns pesquisadores.
- métodos *A Posteriori* - segundo Miettinen (1999) são métodos que podem também ser chamados de métodos para geração das soluções ótimas de Pareto. A tomada de decisão é feita após a realização da busca das soluções Pareto-ótimas. A busca é feita sem considerar preferências de um objetivo sobre outro, isto é, todos objetivos do POMO possuem a mesma relevância. Segundo Lobato (2008) esta categoria de métodos foi proposta por Goldberg (1989) através do conceito de dominância de Pareto e também de um mecanismo de diversidade por nichos para superar as limitações do VEGA (*Vector Evaluated Genetic Algorithms*) (SCHAFFER, 1984).

# Capítulo 3

## Tratamento de POMO - Métodos *A Priori*

As primeiras aplicações envolvendo problemas de otimização multiobjetivo fizeram uso de formas de tratamento para a transformação do problema multiobjetivo em um problema com objetivo único. Outro fato que deve ser salientado é que, para encontrar a solução do problema multiobjetivo, eram utilizados métodos clássicos de otimização para encontrar soluções dos problemas mono-objetivos. Os métodos heurísticos de otimização começaram a ser usados apenas com o advento de computadores com capacidade de processamento capaz de realizar diversos cálculos e armazenar um grande número de informações. Cabe ressaltar que os métodos clássicos, aliados a uma forma de tratamento do POMO, devem realizar várias simulações consecutivas para obter a Curva de Pareto, porém, os métodos heurísticos, usando o conceito de dominância, são capazes de construir a Curva de Pareto em uma única execução. O fluxograma da Figura 3.1 a seguir, ilustra os passos para resolução de POMO sem a utilização do critério de dominância.

Este capítulo é dedicado à descrição dos principais métodos para tratamento de POMO existentes na literatura. Será considerado o problema de otimização multiobjetivo definido pelas Eq.2.1 e 2.2. Para a definição dos métodos neste capítulo, o POMO será

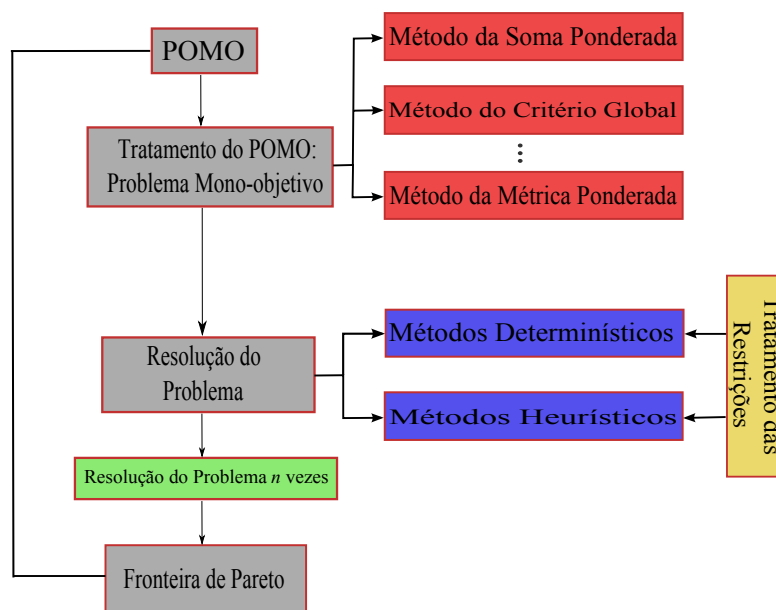


Figura 3.1: Fluxograma de passos para resolução de POMO sem o uso do critério de dominância de Pareto. Adaptado de (LOBATO, 2008).

considerado irrestrito. Metodologias para tratamento de restrições e Métodos Clássicos e Heurísticos de otimização serão descritos com detalhes nos capítulos posteriores desta tese. A demonstração dos teoremas deste capítulo podem ser encontradas em Miettinen (1999).

### 3.1 Método da Soma Ponderada

O Método da Soma Ponderada, do inglês *Weighted Sum Method*, é sem dúvida o mais popular entre os métodos *a Priori* e tem recebido grande atenção dos pesquisadores devido à sua simplicidade conceitual. Baseia-se na transformação do problema multiobjetivo original em um problema mono-objetivo. Cada objetivo é multiplicado por um peso e então minimiza-se a soma ponderada dos objetivos. Assume-se que os pesos  $w_i$ ,  $i = 1, \dots, m$ , são números reais positivos satisfazendo  $\sum_{i=1}^m w_i = 1$ . Mais precisamente, o problema mono-objetivo obtido pela soma ponderada dos objetivos, possui a seguinte



formulação:

$$\text{minimizar } f(x) = \sum_{i=1}^m w_i f_i(x) \quad (3.1)$$

Oliveira (2005) em sua dissertação de mestrado utilizou o método da soma ponderada para a realização de simulações envolvendo problemas de usinagem de metais, otimização de frequências em sistemas massa-mola e em um problema de despacho econômico e ambiental. Sampaio (2011) realizou um estudo teórico e de simulações computacionais envolvendo o método da soma ponderada. Sadek et al. (1989) utilizou o método da soma ponderada para gerar soluções Pareto ótimas para um problema de controle ótimo de uma viga com amortecimento.

Um ponto negativo desta metodologia é que a solução obtida depende fortemente das ponderações (pesos) utilizados. Outros valores adotados para os pesos conduzem a uma solução diferente. Sendo assim um projetista deve realizar diversas simulações, para diferentes pesos, e escolher uma melhor solução com base em sua experiência e conhecimento da realidade do problema, porém a variação continuada dos pesos não garante soluções bem distribuídas sobre a curva solução de Pareto (STADLER, 1995; MESSAC, 1996; DAS; DENNIS, 1997). Um algoritmo para gerar automaticamente diferentes valores para os pesos em problemas convexos (não lineares) foi proposto por Cabalero et al. (1997). Segundo Osyczka (1984) caso deseje que os pesos possam refletir mais fielmente a importância dos objetivos, cada objetivo deve ser expresso na mesma unidade de medida dos outros. Para adimensionalizar os objetivos, reescreve-se o problema 3.1, da seguinte forma:

$$\text{minimizar } f(x) = \sum_{i=1}^m w_i f_i(x) c_i \quad (3.2)$$

onde os  $c_i$  são constantes normalizadoras de cada função e que, segundo Osyczka (1984), obtém-se os melhores resultados quando  $c_i = 1/f_i^*$ , onde  $f_i^*$  é o valor ótimo da  $i$ -ésima função objetivo.

A seguir serão apresentados os principais resultados que relacionam as soluções dos problemas 3.1 e 2.1.

**Teorema 3.1.** *Toda solução do problema 3.1 é Pareto ótima para o problema 2.1.*

*Demonstração.* Ver em Miettinen (1999). □

Quando se assume a hipótese do problema ser convexo, isto é, quando o espaço de objetivos for um conjunto convexo, vale o seguinte resultado.

**Teorema 3.2.** *Considere o problema multiobjetivo de otimização dado pela Eq. 2.1 e suponha que o mesmo seja convexo. Se  $x^*$  é uma solução Pareto ótima então existe um vetor de pesos  $w$  com*

$$w_i \geq 0, \quad i = 1, \dots, m \quad \text{e} \quad \sum_{i=1}^m w_i = 1$$

*tal que  $x^*$  é solução do problema de otimização 3.1.*

De acordo com o teorema anterior, toda solução ótima de Pareto pode ser encontrada pelo método da soma ponderada, caso o problema seja convexo. Uma formulação equivalente, porém com pesos definidos por funções trigonométricas, para o método da soma ponderada é apresentada pelos autores Das e Dennis (1997). Esta formulação pode ser usada para ilustrar geometricamente que nem todas soluções ótimas de Pareto podem ser encontradas pelo método da soma ponderada caso o problema seja não convexo.

### 3.1.1 Interpretação Geométrica

De acordo com os autores Osyczka (1984) e Das e Dennis (1997) o método da soma ponderada é um método específico para abordagens em problemas convexos. Caso o problema seja não convexo pode-se ter soluções ótimas de Pareto que não sejam encontradas pelo método da soma ponderada. Serão apresentadas a seguir a definição de função convexa e uma ilustração para o caso convexo e não convexo.

**Definição 3.3.** *Seja  $f : D_f \rightarrow \mathbb{R}$  uma função real onde  $D_f \subset \mathbb{R}^n$ . Diz-se que  $f$  é uma função convexa quando, para todo  $t \in [0, 1]$ , é verdadeira a desigualdade*

$$f((1-t)x + ty) \leq (1-t)f(x) + tf(y), \quad \forall x, y \in D_f. \quad (3.3)$$

Um conjunto  $D \subset \mathbb{R}^n$  é convexo se dados quaisquer pontos  $x, y \in D$  o segmento de reta  $(1-t)x + ty$ ,  $t \in [0, 1]$ , que liga  $x$  e  $y$  está totalmente contido em  $D$ . Pode ser provado que o espaço de objetivos  $\mathcal{F}$  é convexo se e somente se as funções que compõe o POMO forem convexas.

Observe a Figura 3.2, a seguir. Ela ilustra a aplicação do método da soma ponderada para os casos convexo e não convexo. A Figura 3.2 (a) ilustra o caso convexo onde foi tomada uma reta  $r$  de inclinação  $m_r = -w_1/w_2$ . A minimização do problema 3.1 pode ser obtida através do movimento de  $r$  (sempre paralelamente) em direção a origem, porém mantendo a intersecção entre a reta  $r$  e o espaço de objetivos  $\mathcal{F}$ . Sendo assim o ponto  $P$  obtido através do processo descrito acima é a solução ótima, para estes valores de  $w_1$  e  $w_2$ , do problema 3.1. Observando a Figura 3.2 (b), e realizando o mesmo processo descrito anteriormente, pode-se observar que a solução  $A$  (que é Pareto ótima) não pode ser obtida através da minimização do problema 3.1, independente dos valores de  $w_1$  e de  $w_2$  que se possa escolher. Observa-se também que nenhuma solução Pareto ótima que esteja sobre

a curva  $P_{front}$ , compreendida entre  $C$  e  $D$ , pode ser obtida através da minimização do problema 3.1.

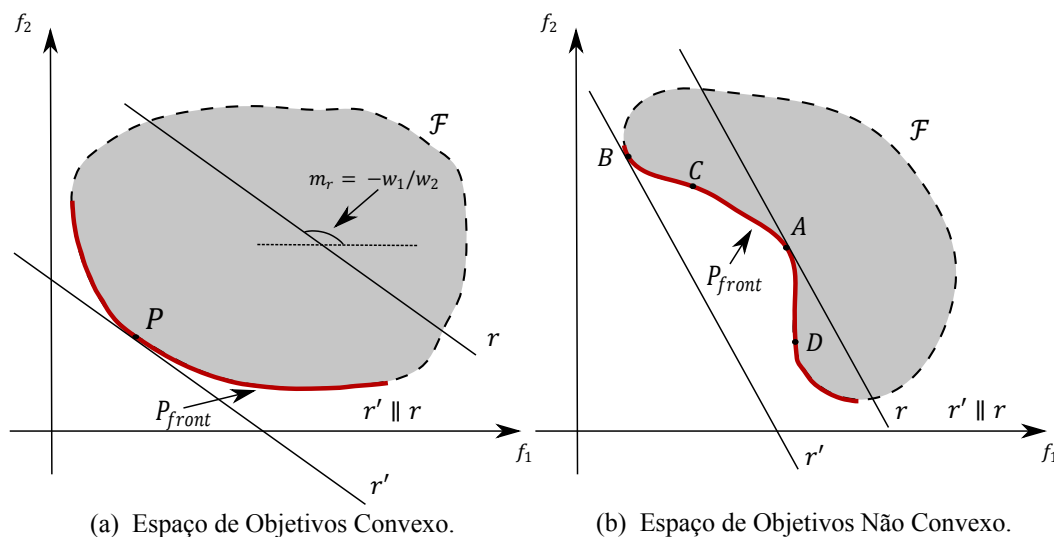


Figura 3.2: Método da soma ponderada: casos convexo e não convexo.

Apesar de ser um método de tratamento de POMO bastante utilizado, o método da soma ponderada possui algumas limitações. Conforme foi ilustrado e comentado anteriormente, pontos ótimos de Pareto em regiões não convexas não são encontrados pelo método (DAS;DENNIS, 1997; MIETTINEN, 2008; SAMPAIO, 2011). Outro fator negativo do método da soma ponderada é o fato de que mesmo para o caso convexo e com várias simulações, para diversos valores dos pesos, não é possível garantir uma boa distribuição das soluções sobre a Curva de Pareto ( $P_{front}$ ).

## 3.2 Método de Otimização Hierárquico

O Método de Otimização Hierárquico foi proposto por Walz (1967) e, conforme o próprio nome diz, consiste na hierarquização dos objetivos de acordo com um critério de importância de cada objetivo. Considere um POMO com os objetivos já hierarquizados

dados por  $f_1(x), f_2(x), \dots, f_m(x)$ . O método de otimização hierárquico consiste em minimizar, na ordem de prioridades pre-estabelecida, cada função objetivo separadamente, adicionando em cada passo uma nova restrição escrita em função de um acréscimo  $\xi_j$  estabelecido a priori.

Assim, a idéia geral do método de otimização hierárquico, pode ser escrita da seguinte forma:

- Encontra-se o ponto de mínimo da primeira função objetivo, ou seja, obtem-se  $x_1^*$ , tal que

$$x_1^* = \min f_1(x), \quad x \in \mathbb{S} = \text{Espaço de Objetivos}$$

- Repetir o processo anterior para  $i = 2, \dots, m$ , isto é, obter  $x_i^*$  solução do problema

$$x_i^* = \min f_i(x), \quad x \in \mathbb{S} = \text{Espaço de Objetivos}$$

com as seguintes restrições:

$$f_{j-1}(x) \leq \left(1 \pm \frac{\xi_{j-1}}{100}\right) f_{j-1}(x_{j-1}^*), \quad j = 2, \dots, i.$$

onde os coeficientes percentuais  $\xi_i$  são acréscimos, se o problema for de minimização, e decréscimos para problemas de maximização.

O método de otimização hierárquico exige, em cada passo, que o projetista entre com os valores dos coeficientes  $\xi_i$  com base no mínimo obtido no passo anterior. Conforme os autores Walz (1967) e Osyczka (1978), um ponto negativo deste método é o fato da qualidade solução final ser extremamente dependente das escolhas dos coeficientes  $\xi_i$ .

### 3.3 Método $\xi$ -Restrito

O método  $\xi$ -restrito foi inicialmente proposto por Haimes et al. (1971). Existem várias versões do método com o objetivo de se encontrar soluções mais eficientes (HAIMES;HALL, 1974; OPPENHEINER, 1978; NULSSEALM;TAVALAGE, 1980; SAKAWA, 1981). Este método também é conhecido na literatura como método de negociação (*Trade-off Method*). Geralmente um método é enquadrado como método negociação se a busca de soluções for guiada pela troca do valor da função objetivo por outro valor da função(OLIVEIRA, 2005). Neste método uma função objetivo é selecionada para ser otimizada e todas outras funções objetivo são convertidas em restrições definindo um limite superior para cada um dos objetivos. A formulação matemática do método  $\xi$ -restrito pode ser escrita na seguinte forma:

$$\text{minimizar } f_i(x) \quad i \in \{1, 2, \dots, m\} \quad (3.4)$$

$$\text{sujeito a } f_j(x) \leq \xi_j \quad j = 1, \dots, m \quad j \neq i \quad (3.5)$$

$$x \in \mathbb{S}$$

Uma formulação alternativa para o método  $\xi$ -restrito é proposto por Lin (1976). Nela o autor propõe restrições de igualdade no lugar das restrições de desigualdade acima propostas.

Segundo Lobato (2008) o procedimento do método  $\xi$ -restrito, descrito acima, deve ser repetido para diversos valores de  $\xi_j$  até que uma solução satisfatória seja encontrada. Osyczka (1984) propõe uma metodologia para auxiliar na escolha dos valores  $\xi_j$ . Em sua metodologia cada função objetivo é minimizada individualmente, obtendo os valores ótimos, denominados por,  $f_i^*$ ,  $i = 1, 2, \dots, m$ . A partir dos valores ótimos calculados,

reescreve-se as restrições descritas na formulação do método  $\xi$ -restrito, adicionando aos valores ótimos uma pequena perturbação, isto é, reescreve-se da seguinte forma:

$$f_j(x) \leq f_j^* + \Delta f_j, \quad j = 1, 2, \dots, m \text{ e } j \neq i.$$

A Figura 3.3 ilustra, para um caso bi-objetivo, a aplicação do método  $\xi$ -restrito para três valores diferentes de  $\xi$ . Inicialmente pode-se observar que valores distintos de  $\xi$  podem conduzir a mesma solução e não é certo que para qualquer valor de  $\xi$  que o método produzirá alguma solução. O ponto  $A$  é a solução obtida do seguinte problema  $\min f_1(x)$  sujeito a  $f_2(x) \leq \xi_1$ . O ponto  $B$  é a solução obtida pelo problema  $\min f_1(x)$  sujeito a  $f_2(x) \leq \xi_2$  e o ponto  $C$  é a solução obtida do seguinte problema  $\min f_1(x)$  sujeito a  $f_2(x) \leq \xi_3$ . Observa-se também que, ao contrário do método da soma ponderada, que o método  $\xi$ -restrito produz qualquer solução ótima de Pareto mesmo o problema não sendo convexo.

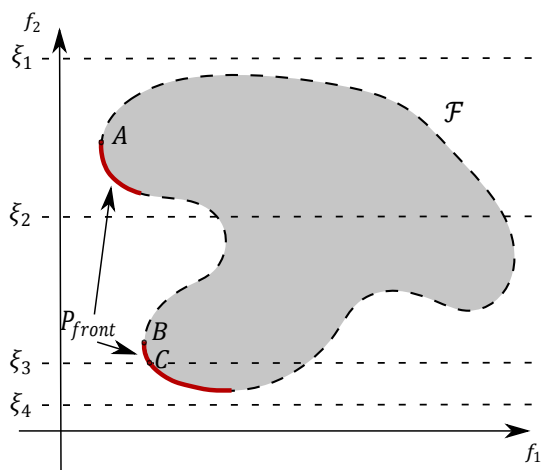


Figura 3.3: Método  $\xi$ -restrito para um problema com dois objetivos.

A seguir serão apresentados os principais resultados que relacionam as soluções dos problemas 3.4 e 2.1.

**Teorema 3.4.** *Toda solução  $x^*$  do problema  $\xi$ -restrito é fracamente Pareto ótima.*

O conteúdo do teorema anterior é que, independente da escolha dos limites superiores  $\xi_j$ , toda solução do problema  $\xi$ -restrito é pelo menos fracamente Pareto ótima.

**Teorema 3.5.** *Um vetor  $x^*$  do espaço de projetos  $\mathbb{S}$  é uma solução Pareto ótima se e somente se é uma solução do problema  $\xi$ -restrito, para todo  $i = 1, 2, \dots, m$ , onde  $\xi_j = f_j(x^*)$  para todo  $j = 1, 2, \dots, m$ .*

*Demonstração.* Ver em Miettinen (1999). □

O teorema anterior diz que toda solução Pareto ótima  $x^*$ , também é solução do problema  $\xi$ -restrito com os limites superiores dados por  $\xi_j = f_j(x^*)$ . Mais ainda, se  $x^*$  for solução do problema  $\xi$ -restrito, com  $\xi_j = f_j(x^*)$ , então ela também será Pareto ótima. Assim sendo, há uma conexão entre as soluções Pareto ótimas e as soluções de problemas  $\xi$ -restritos. O próximo teorema afirma que se  $x^* \in \mathbb{S}$  é *solução única* do problema  $\xi$ -restrito para todo  $l$  com  $\xi_j = f_j(x^*)$ ,  $j = 1, 2, \dots, m$ ,  $l \neq j$ , então necessariamente também será Pareto ótima.

**Teorema 3.6.** *Um vetor  $x^* \in \mathbb{S}$  é Pareto ótimo se é solução única do problema  $\xi$ -restrito para algum  $l$  com  $\xi_j = f_j(x^*)$ ,  $j = 1, 2, \dots, m$ ,  $l \neq j$ .*

### 3.3.1 Conexão com o Método da Soma Ponderada

A seguir será apresentado um resultado que mostra a ligação existente entre o método da soma ponderada e o método  $\xi$ -restrito.

**Teorema 3.7.** *Seja  $x^* \in \mathbb{S}$  uma solução do problema 3.1 e  $w \in \mathbb{R}^m$  seu correspondente vetor de pesos. Então:*



- i) se  $w_i > 0$ ,  $x^*$  é a solução do problema  $\xi$ -restrito para  $f_i(x)$  como função objetivo e  $\xi_j = f_j(x^*)$ ,  $j = 1, 2, \dots, m$ ,  $j \neq i$ ; ou
- ii) se  $x^*$  for a solução única do problema 3.1, então  $x^*$  é a solução do problema  $x_i$ -restrito, onde  $\xi_j = f_j(x^*)$  para  $j = 1, 2, \dots, m$ ,  $j \neq i$ , para todo  $f_i(x)$ ,  $i = 1, 2, \dots, m$ , como função objetivo.

### 3.4 Método Híbrido

É um método que combina os métodos da Soma Ponderada e  $\xi$ -Restrito. Este método foi descrito inicialmente por Corley (1980) e Wendell e Lee (1977) de formas pouco diferentes uma da outra. O nome Método Híbrido foi introduzido em Chankong e Haimes (1983a,b). A expressão matemática do Método Híbrido pode ser escrita da seguinte forma:

$$\text{minimizar } \sum_{i=1}^m w_i f_i(x) \quad (3.6)$$

$$\begin{aligned} \text{sujeito a } f_j(x) &\leq \xi_j & j = 1, 2, \dots, m \\ x &\in \mathbb{S}, \quad w_i > 0, \quad i = 1, 2, \dots, m \end{aligned} \quad (3.7)$$

Note que o problema 3.6 é igual ao problema 3.4 caso se tenha  $w_i = 1$  e  $w_j = 0$ ,  $j = 1, 2, \dots, m$ ,  $j \neq i$ . A seguir é apresentado um resultado que relaciona as soluções do problema 3.6 com soluções Pareto ótimas do POMO original dado pela Eq.2.1.

**Teorema 3.8.** *Toda solução do Problema Híbrido, dado pela Eq.3.6, é Pareto ótima para qualquer vetor de limitantes superiores  $\xi \in \mathbb{R}^m$ . Por outro lado, se  $x^*$  for uma solução Pareto ótima, então é solução do problema híbrido para  $\xi_j = f_j(x^*)$ ,  $j = 1, 2, \dots, m$ .*

### 3.5 Método do Critério Global

O Método do Critério Global também é frequentemente denominado, na literatura corrente, como *Compromise Programming* (YU, 1973; ZELENY, 1973). Neste método minimiza-se a distância entre o vetor de objetivos, sem sair da região viável, isto é, do espaço de projetos; e algum ponto de referência. O projetista deve determinar qual o ponto (solução) de referência e qual métrica a ser utilizada no método.

Com o objetivo de descrever o Método do Critério Global, utiliza-se o vetor de referência como sendo a solução ideal  $f^*$  (vetor dos valores mínimos para cada objetivo) e métricas  $L_p$  para medir a distância. Sendo assim, o  $L_p$ -problema a ser resolvido é dado por:

$$\begin{aligned} &\text{minimizar} \left( \sum_{i=1}^m |f_i(x) - f_i^*|^p \right)^{\frac{1}{p}} \\ &\text{sujeito a } x \in \mathbb{S} \end{aligned} \tag{3.8}$$

Pela definição de solução ideal, sabe-se que  $f_i(x) \geq f_i^*$ , assim não é necessário utilizar o módulo na definição do objetivo do método do critério global descrito acima. Entretanto, em vários casos, não se tem conhecimento sobre a solução ideal e então o método pode utilizar outro ponto de referência, cuidadosamente selecionado, em seu lugar. Sendo assim mantém-se o módulo para incluir o caso onde não se sabe qual é a solução ideal. Pontos de referências pessimistas, no que diz respeito ao problema de otimização, devem ser evitados uma vez que o método não procura soluções melhores do que a solução de referência.

O expoente  $1/p$  pode ser suprimido da definição. Problemas com ou sem o expoente  $1/p$  são equivalentes, para  $1 \leq p < \infty$ , uma vez que o problema  $L_p$ , definido pela Eq.3.8, é uma função decrescente do problema original quando o expoente é eliminado. Se  $p = \infty$  a

métrica é também chamada de *Métrica de Tchebycheff* e o  $L_\infty$  ou problema de Tchebycheff possui a seguinte forma:

$$\begin{aligned} & \text{minimizar} \quad \underbrace{\max}_{i=1,2,\dots,m} |f_i(x) - f_i^*| & (3.9) \\ & \text{sujeito a} \quad x \in \mathbb{S} \end{aligned}$$

Uma variação para o problema de Tchebycheff, proposta por Osyczka (1989, 1992), é expressa por:

$$\begin{aligned} & \text{minimizar} \quad \underbrace{\max}_{i=1,2,\dots,m} \left[ \max \left\{ \left| \frac{f_i(x) - f_i^*}{f_i^*} \right|, \left| \frac{f_i(x) - f_i^*}{f_i(x)} \right| \right\} \right] & (3.10) \\ & \text{sujeito a} \quad x \in \mathbb{S} \end{aligned}$$

Note que o problema 3.8 é não diferenciável devido ao módulo que aparece em sua definição. Porém o mesmo pode ser transformado em um problema diferenciável (caso os objetivos sejam diferenciáveis) através do seguinte problema equivalente:

$$\begin{aligned} & \text{minimizar} \quad \alpha & (3.11) \\ & \text{sujeito a} \quad \alpha \geq f_i(x) - f_i^*, \quad i = 1, 2, \dots, m \\ & \quad \quad \quad x \in \mathbb{S} \end{aligned}$$

Boyчук e Ovchinnikov (1973) e Salukevadze (1974) utilizaram, no método do Critério Global, os valores  $p = 1$  e  $p = 2$ . Segundo estes autores e Miettinen (1999) as soluções encontradas dependem grandemente do valor de  $p$  e em alguns casos pode-se obter soluções inaceitáveis para o projeto. Na grande maioria das situações são escolhidos os valores,  $p = 1, 2$  ou  $p = \infty$ . A Figura 3.4 a seguir, ilustra diferenças entre as soluções obtidas pelo

método do critério global para os valores  $p = 1, 2$  e  $p = \infty$ .

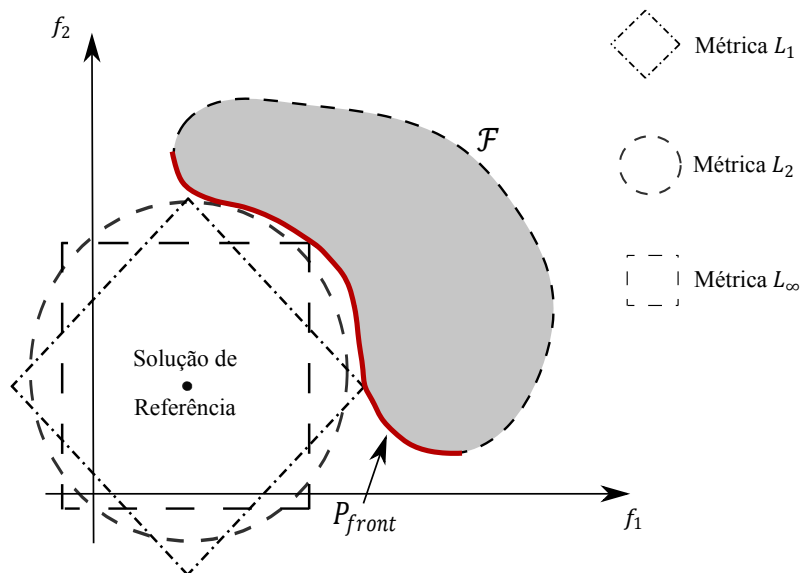


Figura 3.4: Diferenças entre métricas no método do critério global.

A seguir será apresentado um resultado que garante que toda solução do problema 3.8 é um ponto da Curva de Pareto, solução do POMO definido pela Eq. 2.1.

**Teorema 3.9.** *Toda solução do problema  $L_p$  ( $1 \leq p \leq \infty$ ) é solução do problema 2.1, ou seja, é Pareto ótima.*

Yu (1973) mostra em seu artigo que se o espaço de objetivos  $\mathcal{F}$  for convexo, então a solução do problema 3.8 é única.

### 3.6 Método da Métrica Ponderada

Este método é um método híbrido envolvendo os métodos apresentados nas seções 3.1 e 3.5, ou seja, entre os métodos da Soma Ponderada e o Método do Critério Global (Métricas  $L_p$ ) respectivamente. No método 3.1 variando os pesos obtêm-se diferentes soluções Pareto ótimas, enquanto no método 3.5, para obter diferentes soluções, deve-se mudar a solução

de referência e/ou o valor de  $p$ . Sendo assim, no método da Métrica Ponderada, é possível obter diferentes soluções Pareto ótimas através da variação tanto dos pesos quanto da solução de referência e/ou da métrica  $L_p$ . O método a ser descrito nesta seção é também denominado na literatura como Otimização de Compromisso (*Compromise Optimization*) (ZELENY, 1973). O nome Método da Métrica Ponderada é utilizado por Miettinen (1999) em seu livro.

Assume-se, por hipótese, que os pesos satisfazem  $w_i \geq 0$  para todo  $i = 1, 2, \dots, m$  e que  $\sum_{i=1}^m w_i = 1$ . A formulação matemática do método da Métrica Ponderada é expressa pelo seguinte problema:

$$\begin{aligned} & \text{minimizar} \quad \left( \sum_{i=1}^m w_i |f_i(x) - f_i^*|^p \right)^{\frac{1}{p}} \\ & \text{sujeito a} \quad x \in \mathbb{S} \end{aligned} \tag{3.12}$$

onde  $1 \leq p \leq \infty$ .

O problema 3.12 é naturalmente não diferenciável, devido ao módulo, como o Método do Critério Global. Porém, como no caso do Método do Critério Global e de forma inteiramente análoga, é possível transformá-lo em um problema diferenciável (caso os objetivos forem diferenciáveis). Será apresentado um resultado sobre quais condições uma solução do Problema 3.12 deve satisfazer para ser ótima de Pareto.

**Teorema 3.10.** *Seja  $x^* \in \mathbb{S}$  uma solução do problema 3.12. Então  $x^*$  é um ponto da Curva de Pareto caso uma das condições sejam verdadeiras:*

- i) a solução  $x^*$  é única; ou*
- ii) todos os pesos são constantes positivas.*

A convexidade em um problema de otimização multiobjetivo é necessária para garantir que toda solução Pareto ótima pode ser encontrada através do Método da Métrica Ponderada (SAWARAGI; NAKAYAMA; TANINO, 1985).

### 3.7 Método do Critério Ponderado Exponencial

Conforme foi mostrado na seção inicial deste capítulo, o método da soma ponderada possui dificuldades de encontrar soluções em regiões não convexas do espaço de objetivos  $\mathcal{F}$ . Objetivando contornar este problema, Athan e Papalambros (1996) propuseram um método denominado Método do Critério Ponderado Exponencial (*Exponential Weighted Criterion*). O método de tratamento de POMO consiste em:

$$\begin{aligned} &\text{minimizar } \sum_{i=1}^m (e^{pw_i} - 1) e^{pf_i(x)} && (3.13) \\ &\text{sujeito a } x \in \mathbb{S} \end{aligned}$$

A seguir é apresentado um resultado que relaciona soluções do problema do Critério Ponderado Exponencial 3.13 com soluções Pareto ótimas do POMO 2.1.

**Teorema 3.11.** *Seja  $x^* \in \mathbb{S}$  uma Pareto ótima do POMO 2.1. Então  $x^*$  é solução do problema 3.13.*

### 3.8 Método do Produto Ponderado (*Weighted Product Method*)

Proposto por Bridgman (1922), o Método do Produto Ponderado, como o método da soma ponderada, consiste em atribuir pesos para cada objetivo de acordo com sua ordem de importância. Porém, no produto ponderado, os pesos aparecem como potências e os

---

objetivos multiplicados uns pelos outros. A formulação matemática é expressa através do problema:

$$\begin{aligned} & \text{minimizar} \quad \prod_{i=1}^m |f_i|^{w_i}(x) \\ & \text{sujeito a} \quad x \in \mathbb{S} \end{aligned} \tag{3.14}$$

A utilização do módulo se dá apenas para garantir que, para qualquer escolha dos pesos  $w_i$ 's, a potenciação possa fazer sentido.

Marler e Arora (2004), em seu artigo, realizaram uma ampla pesquisa das principais estratégias para resolução de POMO. Tanto o artigo dos autores citados quanto suas referências constituem uma excelente fonte de pesquisa de estratégias de resolução de POMO.

# Capítulo 4

## Métodos Clássicos de Otimização

Neste capítulo serão descritos os principais métodos clássicos de otimização da literatura bem como métodos de tratamento de restrições. Os métodos serão definidos para problemas mono-objetivos irrestritos, uma vez que, problemas de otimização multiobjetivo podem ser transformados em problemas mono-objetivos através de técnicas descritas com detalhes no capítulo anterior e as restrições são incorporadas ao objetivo através de técnicas definidas no final deste capítulo. Assim, durante este capítulo e na definição dos métodos, o problema geral de otimização considerado é dado por

$$\begin{aligned} &\text{minimizar } f(x) && (4.1) \\ &\text{sujeito a } x \in \mathbb{S} \subset \mathbb{R}^n \end{aligned}$$

### 4.1 Noções de Convergência

Qualquer método proposto para resolver o problema de otimização 4.1 realiza avaliação da função objetivo e muitos necessitam de avaliações das derivadas desta função. Um método



é chamado de **método sequencial**, quando cada ponto se define a partir de uma recorrência (informações obtidas) aos pontos anteriores. Segundo Izmailov e Solodov (2007) a maioria dos métodos clássicos de otimização são do tipo sequenciais. Há também os métodos ditos **métodos passivos** onde os pontos são obtidos através de uma estratégia pré-definida e não leva em consideração informações obtidas ao longo do processo iterativo. Os métodos sequenciais são mais eficientes do que os métodos passivos (IZMAILOV;SOLODOV, 2007).

Um método sequencial gera uma sequência  $\{x_k\}_{k \in \mathbb{N}}$  em  $\mathbb{S}$  que é chamada de *aproximações* à solução do problema, ou de *iterandos* do método. A geração do ponto  $x_{k+1}$  a partir de  $x_k$  é chamada de *iteração* do método e os detalhes de como se obtém  $x_{k+1}$  a partir de  $x_k$  constituem a natureza do método. Frequentemente, métodos são descritos em forma de um processo iterativo da seguinte forma:

$$x_{k+1} = \varphi(x_k), \quad k = 0, 1, 2, \dots, \quad (4.2)$$

onde  $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}^n$

Naturalmente supõe-se que o operador  $\varphi$  é conhecido e normalmente depende de avaliações da função objetivo e/ou da sua derivada nos pontos da sequência gerada. Fixada uma escolha de um semente inicial  $x_0 \in \mathbb{S}$ , então o processo iterativo 4.2 define uma única sequência de soluções  $\{x_k\}_{k \in \mathbb{N}}$  dada por

$$x_1 = \varphi(x_0), x_2 = \varphi(x_1), x_3 = \varphi(x_2), \dots, x_n = \varphi(x_{n-1}), \dots, \quad (4.3)$$

O esquema descrito em 4.2 é dito sem memória. Há também processos iterativos com memória, onde a iteração  $x_{k+1}$  depende não somente de  $x_k$ , mas também de outros pontos

da sequência. Um exemplo de processos iterativos com memória é a famosa sequência de Fibonacci, onde os pontos  $x_0$  e  $x_1$  são 0 e 1, respectivamente, e cada termo posterior é dado pela soma dos dois anteriores (SIGLER, 2002).

A ordem de um método é a máxima ordem das derivadas da função objetivo do problema de otimização. Por exemplo, um método que se baseia apenas em avaliações da função objetivo é chamado de **método de ordem zero**, pois não faz uso da derivada. Um método que utiliza derivada primeira (segunda) do objetivo é dito **método de primeira (segunda) ordem**.

Há métodos de convergência finita, isto é, a sequência dos iterados  $x_k$  convergem para a solução do problema de otimização 4.1 para um valor de  $k$  finito. Porém algoritmos com esta propriedade existem apenas para problemas bastante simples, como problemas de programação linear e de programação quadrática. Em geral, as aplicações em engenharia conduzem a problemas não lineares bastante complexos e os métodos para estes problemas são de convergência assintótica, isto é, quanto maior o número de iterações mais próxima a sequência dos iterados estará da solução do problema de otimização. Em termos matemáticos a convergência assintótica pode ser escrita da seguinte forma:

**Definição 4.1.** *Seja  $x^*$  uma solução do problema de otimização 4.1. Diz-se que a sequência  $\{x_k\}_{k \in \mathbb{N}}$ , converge para  $x^*$ , se para todo  $\epsilon > 0$  dado, existe um índice  $k_0$  tal que*

$$\|x_k - x^*\| < \epsilon, \text{ para todo } k \geq k_0. \quad (4.4)$$

A notação utilizada para denotar que a sequência  $\{x_k\}_{k \in \mathbb{N}}$  converge para a solução  $x^*$  é

$$\lim_{k \rightarrow \infty} x_k = x^*, \text{ ou simplesmente, } x_k \rightarrow x^*. \quad (4.5)$$

Pode-se ter também convergência em relação à função objetivo, isto é,  $f(x_k) \rightarrow f^*$  onde  $f^*$  é o valor ótimo do problema. Porém este tipo de convergência não garante que a trajetória  $\{x_k\}$  do método convirja para uma solução ótima  $x^*$ .

A maioria dos métodos clássicos de otimização são métodos de convergência local, ou seja, a convergência é garantida (sob hipóteses matemáticas) apenas se a escolha de um ponto inicial  $x_0$  se der em uma vizinhança de uma solução  $x^*$ . Izmailov e Solodov (2007) afirmam que todo método de convergência local deve ser pensado em combinação com algum outro método capaz de fornecer um ponto inicial que seja adequado para uma aplicação bem sucedida do método local (*globalização da convergência*).

## 4.2 Taxas de Convergência

Segundo Izmailov e Solodov (2007) para a análise e avaliação de métodos com convergência assintótica, a questão sobre a *taxa de convergência* é uma das mais importantes, pois é um dos principais indicadores de eficiência em um método iterativo. Suponha que um dado método fornece uma sequência  $\{x_k\}$  convergente para uma solução  $x^*$ , isto é,  $x_k \rightarrow x^*$  ( $k \rightarrow \infty$ ). Resultados sobre a taxa de convergência fornecem garantias na redução da distância entre  $x_{k+1}$  e  $x^*$  em relação à distância entre  $x_k$  e  $x^*$ , para valores de  $k$  suficientemente grandes. Nas definições abaixo será assumida a hipótese de  $\{x_k\}$  ser uma sequência gerada por algum método e que esta converge para uma solução  $x^*$  do problema de otimização 4.1.

**Definição 4.2.** *Diz-se que a sequência  $\{x_k\}$  converge linearmente para  $x^*$  se existe  $c \in (0, 1)$  tal que*

$$\|x_{k+1} - x^*\| \leq c \|x_k - x^*\| \quad (4.6)$$

para todo  $k \geq k_0$ , onde  $k_0$  é um índice suficientemente grande.

**Definição 4.3.** Diz-se que a sequência  $\{x_k\}$  converge quadraticamente para  $x^*$  se existe  $c \in [0, 1)$  tal que

$$\|x_{k+1} - x^*\| \leq c\|x_k - x^*\|^2 \quad (4.7)$$

para todo  $k \geq k_0$ , onde  $k_0$  é um índice suficientemente grande.

**Definição 4.4.** (Erro Cometido) O erro cometido na  $k$ -ésima iteração é definido por:

$$e_k = \|x_k - x^*\|, k = 0, 1, 2, \dots, \quad (4.8)$$

Observe que, nas definições acima, muda-se apenas o expoente do módulo do lado direito. Uma convergência mais lenta que a linear é dita *sublinear* enquanto que uma convergência maior que a linear é dita *superlinear*. Pode ser provado que, em um processo iterativo, se a convergência for linear, então para dobrar o número de casas decimais exatas é necessário dobrar o número de iterações. Por outro lado, se a convergência for quadrática, então para dobrar o número de casas decimais exatas é suficiente apenas mais uma iteração (BURDEN;FAIRES, 2005).

Mais informações sobre diferentes estimativas e taxas de convergência podem ser encontradas em Bertsekas (1999) e Ortega e Rheinboldt (1970). Segundo Izmailov e Solodov (2007) estimativas de taxa de convergência fornecem informação útil para comparação qualitativa entre métodos diferentes. Porém, ainda segundo Izmailov e Solodov (2007), a taxa de convergência não é a única característica relevante para comparação entre métodos distintos. É importante considerar o custo computacional envolvido em uma iteração. Por exemplo, um método com taxa de convergência maior, pode ser mais lento na prática, isto

é, levar um tempo maior para resolver um problema, se o custo computacional da iteração dele é mais alto.

### 4.2.1 Critérios de Parada

Mesmo num método de convergência assintótica, sabe-se que nenhum processo computacional pode ser infinito. Assim, qualquer método de convergência assintótica, que será implementado computacionalmente, necessita de algum *critério ou regra de parada*. Segundo Varderplaats (1999) e Edgar et al. (2001) a característica iterativa dos métodos de otimização constituem uma vantagem na implementação computacional, pois podem passar a ser executadas de forma automática, até que algum critério de parada seja satisfeito.

Segundo Izmailov e Solodov (2007) os critérios de parada mais confiáveis são exatamente os que são baseados em estimativas da taxa de convergência e em outras propriedades conhecidas (por exemplo, estimativas da distância ao conjunto de soluções). Quando a taxa de convergência é conhecida, então ela pode ser usada para estimar o número de iterações que são necessárias para se ter uma solução aproximada com uma precisão estabelecida à priori. Porém, encontrar esta estimativa não é uma tarefa simples (IZMAILOV;SOLODOV, 2007). Na maioria dos casos, as taxas de convergência possuem natureza local, e a quantificação do número de iterações para conseguir um ponto da iteração a partir do qual valha esta taxa não é possível. Outro fator que torna difícil a utilização da taxa de convergência como estimativa para algum processo de parada é que na definição de taxa de convergência aparecem constantes que, quase sempre, são desconhecidas. Sendo assim, a aplicação prática de taxas de convergência para definição de algum critério de parada em termos do número de iterações é algo bastante limitado (IZMAILOV;SOLODOV,2007).

Na prática, se utilizam regras que são menos confiáveis do ponto de vista teóricos

e que podem, porém ser mais facilmente implementadas computacionalmente. Essas regras são baseadas em informações obtidas na sequência dos iterados  $\{x_k\}$ . Comumente, nos processos conhecidos na literatura, se examina o comportamento de uma parte mais recente da sequência  $\{x_k\}$  e/ou da sequência  $\{f(x_k)\}$  e/ou de alguma medida de violação de condições de otimalidade e/ou de um número máximo de iterações realizadas.

Segundo Burden e Faires (2005), Izmailov e Solodov (2007) e Franco (2007) os principais critérios de parada para métodos iterativos são expressos por:

- fixado um número pequeno  $\epsilon > 0$ , o método é parado após a iteração de índice  $k + 1$  se

$$\|x_{k+1} - x_k\| \leq \epsilon \quad (4.9)$$

ou seja, quando o passo do método fica curto, acredita-se não ser possível melhorar a qualidade de aproximação a uma solução do problema de otimização;

- outro critério, que de certo modo é análogo ao anterior, é definido através da seguinte expressão:

$$\|\nabla f(x_{k+1}) - \nabla f(x_k)\| \leq \epsilon \quad (4.10)$$

o entendimento que se tem da expressão acima é que se o gradiente (o oposto do gradiente é a direção de máxima descida), não sofre alterações significativas avaliado nos pontos da sequência  $\{x_k\}$ , gerada por algum método, então não será possível melhorar a solução obtida até o momento;

- há também um critério utilizado com frequência como regra de parada em métodos

de otimização que é dado por:

$$\|\nabla f(x_{k+1})\| \leq \epsilon \quad (4.11)$$

Este critério deve ser usado com bastante cuidado, pois a menos que se tenha uma ideia bastante clara do comportamento da função, pode-se ter dificuldade com a análise da resposta do teste. Por exemplo, considere a seguinte função

$$f(x) = -\frac{x^{-2}}{4} (2 \ln x + 1) \quad (4.12)$$

onde a única raiz da derivada (ponto crítico, que no caso é de mínimo), é a solução  $x^* = 1$ . Observe que, tomando a sequência geométrica  $x_1 = 2, x_2 = 4, x_3 = 8, x_5 = 16, x_6 = 32, \dots$  obtem-se os valores da derivada  $f'(x) = x^{-3} \ln x$ , 0.0866, 0.0217, 0.00406, 0.0006769, 0.0001058,  $\dots$ , respectivamente. Assim quanto mais longe a sequência está de  $x^*$  menor é o valor da derivada  $f'(x)$ .

- uma regra de parada que é bastante utilizada, principalmente em métodos heurísticos de otimização, é o número máximo de iterações. Mesmo que se utilize outra regra, como as dadas nos itens anteriores, é importante utilizar também esta regra, pois, em muitos casos, as regras anteriores podem conduzir a um processo que nunca seja satisfeito.

Segundo Izmailov e Solodov (2007) e Franco (2007) é claro que os critérios apresentados nos itens acima não são completamente confiáveis, pois, em geral, eles não garantem a proximidade da sequência dos iterados  $\{x_k\}$  a uma solução do problema. Há exemplos de sequências  $\{x_k\}$  satisfazendo  $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$  e as mesmas não serem convergentes. Por exemplo, basta considerar as sequências  $\{x_k\} \subset \mathbb{R}$  dadas por  $x_k = \sqrt{k}$  ou  $x_k =$

---

$\sin \sqrt{k}, k = 0, 1, 2, \dots$ . Ainda que não confiáveis, os critérios apresentados são comuns em métodos computacionais de otimização, simplesmente porque não há nenhuma alternativa melhor (IZMAILOV;SOLODOV,2007).

Em geral, em métodos computacionais, utiliza-se, de forma combinada, mais de um critério de parada, sendo que um número máximo de iterações ou de avaliações da função objetivo é um critério implícito para qualquer método de otimização. A combinação de critérios de parada envolve uma hierarquização dos critérios, sendo que, qualquer hierarquização é muito mais uma questão de arte e da experiência do projetista do que uma questão da ciência exata.

### 4.3 Métodos de Otimização Unidimensional

Nesta seção serão apresentados alguns dos métodos de otimização unidimensional mais comuns da literatura. São métodos que, pela sua simplicidade conceitual, não são capazes de resolver a maioria dos problemas complexos em engenharia, porém, são bastante úteis no desenvolvimento de ferramentas teóricas e computacionais mais sofisticadas.

Nas subseções que se seguem serão apresentados métodos unidimensionais para resolver o seguinte problema

$$\begin{aligned} &\text{minimizar } f(x) && (4.13) \\ &\text{sujeito a } x \in [a, b] \end{aligned}$$

onde  $f : [a, b] \rightarrow \mathbb{R}$  é uma função, que para a finalidade de definição dos métodos, apenas contínua em  $[a, b]$ .



### 4.3.1 Método da Comparação de Pontos de Rede

Conforme o próprio nome diz, o Método da Comparação de Pontos de Rede consiste em dividir o domínio da função objetivo  $f$  em  $N$  pontos, avaliar o valor de  $f$  nos pontos criados e então escolher aquele que fornece o menor valor entre todos os pontos criados. Talvez este seja o método de otimização mais simples (simplicidade conceitual) que existe. A partir da escolha da quantidade de pontos  $N$ , a serem criados, o método é estabelecido através da seguinte rotina:

$$\text{Defina} \quad \delta_N = \frac{b - a}{N} \quad (4.14)$$

$$\text{Calcule} \quad x_i^N = (a + i\delta_N) - \frac{\delta_N}{2}, i = 1, 2, \dots, N \quad (4.15)$$

$$\text{Escolha} \quad \bar{x}^* = \underbrace{\min}_{i=1,2,\dots,N} f(x_i^N) \quad (4.16)$$

Observe que este é um método passivo, isto é, não fornece uma sequência de pontos  $\{x_k\}$  que aproxima uma solução  $x^*$  e que não é necessário estabelecer nenhum critério de parada. Para se ter uma solução  $\bar{x}^*$  que aproxime a solução exata  $x^*$  com precisão menor que  $\epsilon$  é suficiente tomar  $N$  tal que  $(b - a)/N < \epsilon$ .

### 4.3.2 Método da Bissecção

Nesta seção será definido outro método de otimização que não depende do uso de derivadas e que gera uma sequência  $\{x_k\}$  convergente (assintoticamente) para um ponto de mínimo da função objetivo. Primeiramente considere a seguinte definição.

**Definição 4.5.** Diz-se que uma função  $f : [a, b] \rightarrow \mathbb{R}$  é unimodal em  $[a, b]$ , quando ela possui um único minimizador global  $x^* \in [a, b]$ , e é estritamente decrescente em  $[a, x^*]$  e estritamente crescente em  $[x^*, b]$ .

De posse da definição anterior, pode-se enunciar um Lema, que é a base do método da Bissecção para problemas unidimensionais de otimização.

**Lema 4.6.** *Sejam  $f : [a, b] \rightarrow \mathbb{R}$  uma função unimodal e  $x^*$  o minimizador global de  $f$  em  $[a, b]$ . Então, para quaisquer pontos  $y, z \in [a, b]$ , com  $y < z$ , vale o seguinte:*

*i) se  $f(y) \leq f(z)$ , então  $x^* \in [a, z]$ ,*

*ii) se  $f(y) \geq f(z)$ , então  $x^* \in [y, b]$ .*

*Demonstração.* Ver em Izmailov e Solodov (2007). □

Como consequência do Lema anterior, comparando os valores da função unimodal  $f$  em dois pontos de  $[a, b]$ , pode-se obter a localização do mínimo global  $x^*$  em um intervalo de menor amplitude. Realizando este procedimento sucessivas vezes, gera-se uma sequência de intervalos, com amplitude sempre menor do que a amplitude anterior. Quando se obter um intervalo  $[a_k, b_k]$  tal que  $b_k - a_k < \epsilon$ , onde  $\epsilon$  é a precisão requerida, é suficiente tomar o ponto médio do intervalo  $[a_k, b_k]$  como aproximação da solução  $x^*$ .

Seja  $f : [a, b] \rightarrow \mathbb{R}$  uma função unimodal. Defina  $a_1 = a, b_1 = b, c_1 = (a + b)/2$  e calcule  $f(c_1)$ . Tome  $k := 1$ . Então, pode-se estabelecer a seguinte rotina para o método da bissecção:

*i) Defina  $y_k = (a_k + c_k)/2$  e calcule  $f(y_k)$ . Se  $f(y_k) \leq f(c_k)$ , definir  $a_{k+1} = a_k, b_{k+1} = c_k, c_{k+1} = y_k$  e passar ao terceiro ítem. Se  $f(y_k) > f(c_k)$ , definir  $z_k = (c_k + b_k)/2$  e calcular  $f(z_k)$ ;*

*ii) Se  $f(c_k) \leq f(z_k)$ , definir  $a_{k+1} = y_k, b_{k+1} = z_k, c_{k+1} = c_k$ . Se  $f(c_k) > f(z_k)$ , definir  $a_{k+1} = c_k, b_{k+1} = b_k, c_{k+1} = z_k$ ,*

*iii) Tomar  $k := k + 1$  e retornar ao primeiro ítem.*

O Lema 4.6 garante que, após a iteração de índice  $k$ , tem-se que  $x^* \in [a_{k+1}, b_{k+1}]$ . Além disso, tem-se que  $c_{k+1} = (a_{k+1} + b_{k+1})/2$  e

$$b_{k+1} - a_{k+1} = \frac{b_k - a_k}{2} = \dots = \frac{1}{2^k}(b - a)$$

Suponha que o número máximo de avaliações da função objetivo seja  $N$  e que  $N$  seja um número ímpar (apenas por conveniência). Neste caso, pode-se realizar, no máximo,  $k = (N - 1)/2$  iterações do método, o que resulta na seguinte estimativa para o erro de aproximação:

$$\|x_{k+1} - x_k\| \leq b_{k+1} - a_{k+1} \leq \frac{1}{2^{(N-1)/2}}(b - a) \approx 0.707^{N-1}(b - a) \quad (4.17)$$

Resulta que, aumentando  $N$ , o erro diminui com taxa geométrica com razão  $q = 0.707$ . A Figura 4.1 a seguir, ilustra a aplicação do método da bissecção para minimização de uma função real.

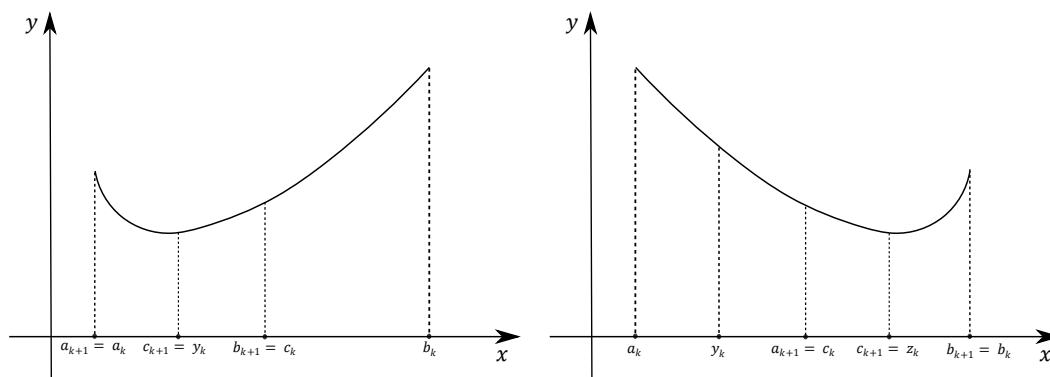


Figura 4.1: Aplicação de uma iteração do método da Bissecção.

Apesar de serem métodos unidimensionais e que sozinhos não são capazes de resolver problemas complexos em engenharia, os métodos a serem descritos são bastante úteis no cálculo do tamanho do passo para o Método do Gradiente ou como é mais comum na

literatura Método de Máxima Descida (LOBATO, 2008; BRANDÃO, 2010).

### 4.3.3 Método da Seção Áurea

O Método da Seção Áurea é um método que consiste em particionar o intervalo  $[a, b]$ , domínio de uma função unimodal, em duas partes tais que a razão entre o comprimento do intervalo inicial  $[a, b]$  com o comprimento da parte maior é igual a razão entre parte maior com o comprimento da parte menor. Suponha que seja escolhido um ponto  $z$  no intervalo  $[a, b]$  e que  $z$  esteja mais próximo de  $a$  do que de  $b$ , conforme ilustra a Fig. 4.2 que se segue.

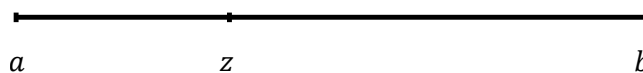


Figura 4.2: Escolha de um ponto  $z$  no intervalo  $[a, b]$ .

Então, considerando as razões entre os segmentos descrita no parágrafo anterior, obtem-se que:

$$\frac{b-a}{b-z} = \frac{b-z}{z-a}. \quad (4.18)$$

A equação quadrática que representa a equação anterior e suas raízes no intervalo  $[a, b]$  são dadas, respectivamente, por:

$$z^2 + (-3b + a)z + (b^2 + ab - a^2) = 0 \quad (4.19)$$

$$y(a, b) = \left( \frac{3 - \sqrt{5}}{2} \right) b + \left( \frac{1 + \sqrt{5}}{2} \right) a \quad (4.20)$$

$$z(a, b) = \left( \frac{3 + \sqrt{5}}{2} \right) b + \left( \frac{1 - \sqrt{5}}{2} \right) a \quad (4.21)$$

Note que a raiz  $y$  é o menor e que a raiz  $z$  é o maior ponto da seção áurea do intervalo  $[a, b]$ . A partição descrita acima possui uma certa relação com a *Sequência de Fibonacci* e, evidentemente, com o *Número Áureo*  $(1 + \sqrt{5})/2$  (IZMAILOV ; SOLODOV, 2007).

A seguir é apresentado o Método da Seção Áurea. Supõe-se que a função  $f$  seja unimodal. Defina  $a_1 = a, b_1 = b, y_1 = y(a, b)$  e  $z_1 = z(a, b)$  e tome  $k := 1$ . Então:

*i)* Se  $f(y_k) \leq f(z_k)$ , definir então  $a_{k+1} = a_k, b_{k+1} = z_k, y_{k+1} = y(a_{k+1}, b_{k+1})$  e  $z_{k+1} = y_k$ . Se  $f(y_k) > f(z_k)$ , definir então  $a_{k+1} = y_k, b_{k+1} = b_k, y_{k+1} = z_k$  e  $z_{k+1} = z(a_{k+1}, b_{k+1})$ ;

*ii)* Tomar  $k := k + 1$  e retornar ao primeiro ítem.

A Figura 4.3 a seguir ilustra a aplicação de uma iteração do Método da Seção Áurea.

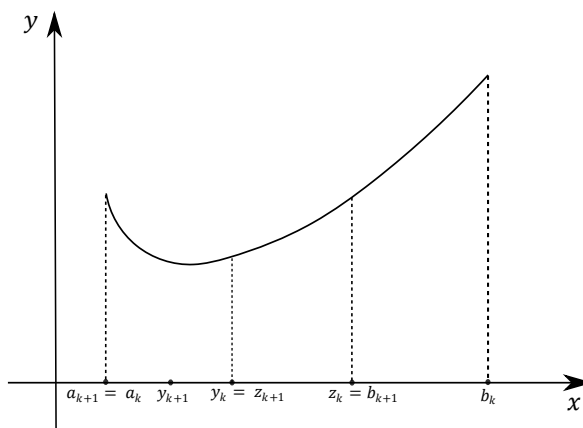


Figura 4.3: Aplicação de uma iteração no método da Seção Áurea.

Será feita uma estimativa sobre a convergência do método baseada na diminuição do intervalo original  $[a, b]$ . Primeiramente observe que  $y_{k+1} = y(a_{k+1}, b_{k+1})$  e  $z_{k+1} = z(a_{k+1}, b_{k+1})$ . Sendo assim, tem-se que  $x^* \in [a_{k+1}, b_{k+1}]$  e vale a seguinte expressão

$$b_{k+1} - a_{k+1} = \frac{(\sqrt{5} - 1)(b_k - a_k)}{2} = \dots = \left( \frac{\sqrt{5} - 1}{2} \right)^k (b - a). \quad (4.22)$$

Na  $k$ -ésima iteração, caso o tamanho do intervalo  $[a_{k+1}, b_{k+1}]$  seja menor que a precisão requerida, toma-se qualquer ponto  $x_{k+1} \in [a_{k+1}, b_{k+1}]$  como aproximação à solução  $x^*$ .

Note que a cada iteração realiza-se apenas uma avaliação da função objetivo, então,  $N$  avaliações da função objetivo permitem realizar  $k - 1$  iterações. Assim, é possível obter a seguinte estimativa de erro para o Método da Seção Áurea:

$$\|x_{k+1} - x^*\| \leq b_{k+1} - a_{k+1} \quad (4.23)$$

$$= \left( \frac{\sqrt{5} - 1}{2} \right)^{N-1} (b - a) \approx 0.618^{N-1} (b - a). \quad (4.24)$$

Portanto, observa-se que no Método da Seção Áurea o erro tende a zero, quando  $N$  tende ao infinito, mais rapidamente do que no Método da Bissecção. Segundo Brandão (2010), outro fato importante, que dá nome ao método, é que a redução do tamanho do intervalo, de uma iteração para outra, é obtida através de uma fração do número áureo. O número áureo aparece em diversas aplicações da matemática em todas as áreas. Algumas curiosidades sobre o número áureo podem ser encontradas em Dunlap (1997) e Livio (2002).

#### 4.3.4 Método de Interpolação Polinomial

A ideia central dos Métodos de Interpolação Polinomial é aproximar a função objetivo  $f$  por um polinômio, usando valores de  $f$  em um certo número de pontos. O ponto que minimiza o polinômio interpolador serve como aproximação seguinte ao minimizador da função objetivo  $f$ . A seguir é apresentada a implementação para a interpolação quadrática.

Considere  $f$  uma função a ser minimizada no intervalo  $[a, b]$ . Para aproximar, por interpolação, a função  $f$  por um polinômio quadrático, é necessário, três pontos no intervalo dado. Defina  $a_1 = a, b_1 = b$  e tome  $y_1$  no interior do intervalo  $[a, b]$ , ou seja,  $a_1 < y_1 < b_1$

e tome  $k := 1$ . A seguir é apresentada a rotina que deve ser implementada no método de interpolação polinomial quadrática.

i) Calcular

$$x_k = \frac{1}{2} \frac{f(a_k)(c_k^2 - b_k^2) + f(b_k)(a_k^2 - c_k^2) + f(c_k)(b_k^2 - a_k^2)}{f(a_k)(c_k - b_k) + f(b_k)(a_k - c_k) + f(c_k)(b_k - a_k)}; \quad (4.25)$$

ii) Se  $x_k > y_k$  e  $f(x_k) < f(b_k)$ , definir então  $a_{k+1} = x_k, y_{k+1} = y_k, b_{k+1} = b_k$ ;

Se  $x_k > y_k$  e  $f(x_k) > f(b_k)$ , definir então  $b_{k+1} = x_k, a_{k+1} = a_k, y_{k+1} = y_k$ ;

Se  $x_k < y_k$  e  $f(x_k) < f(b_k)$ , definir então  $b_{k+1} = x_k, a_{k+1} = a_k, y_{k+1} = y_k$ ;

Se  $x_k < y_k$  e  $f(x_k) > f(b_k)$ , definir então  $a_{k+1} = x_k, y_{k+1} = y_k, b_{k+1} = b_k$ ;

iii) Tomar  $k := k + 1$  e retornar ao primeiro ítem.

## 4.4 Métodos de Descida

Para a descrição dos principais aspectos dos métodos de descida será considerado o seguinte problema irrestrito em  $n$  variáveis

$$\begin{aligned} \text{minimizar } f(x) \quad \text{onde } f : \mathbb{R}^n \rightarrow \mathbb{R} \\ x \in \mathbb{R}^n \end{aligned} \quad (4.26)$$

Como o próprio nome diz, os métodos de descida consistem em, dada uma aproximação  $x_k$  da solução do problema 4.26, encontrar uma direção  $d_k$  tal que  $f$  seja decrescente (pelo menos para pequenos passos) e a partir de  $x^k$  *caminhar* na região de decrescimento, seguindo a direção  $d_k$ . Em cada iteração é necessário calcular uma direção de descida  $d_k$  e

o tamanho do passo  $\alpha_k > 0$ . Os valores  $d_k$  e  $\alpha_k$  devem satisfazer a seguinte desigualdade

$$f(x_k + \alpha_k d_k) < f(x_k). \quad (4.27)$$

Assim, o iterando seguinte é obtido através da seguinte fórmula

$$x_{k+1} = x_k + \alpha_k d_k \quad (4.28)$$

Brandão (2010) realiza, em sua dissertação de mestrado, um estudo teórico analítico e experimental de um método de descida. Lobato (2008) descreve sobre um método de descida e apresenta uma descrição analítica e gráfica das condições de otimalidade, denominada condições de Karush-Kuhn-Tucker(KKT), para um problema com restrições de igualdade.

Para descrever sobre métodos de descida é necessário primeiramente saber o que significa direção de descida para, depois, saber como calcular o tamanho do passo na direção de descida. A seguir é apresentada a definição de direção de descida (IZMAILOV;SOLODOV, 2007; LOBATO, 2008; BRANDÃO,2010).

**Definição 4.7.** Diz-se que  $d \in \mathbb{R}^n$  é uma **direção de descida** da função  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  no ponto  $x \in \mathbb{R}^n$ , se existe um  $\epsilon > 0$  tal que

$$f(x + td) < f(x) \quad \forall t \in (0, \epsilon] \quad (4.29)$$

Denota-se por  $\mathcal{D}_f(x)$  o conjunto de todas as direções de descida da função  $f$  no ponto  $x$ .

**Lema 4.8.** Seja  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  uma função diferenciável no ponto  $x \in \mathbb{R}^n$ . Então:

*i)* Para todo  $d \in \mathcal{D}_f(x)$ , tem-se que  $\langle \nabla f(x), d \rangle \leq 0$ ;



ii) Se  $d \in \mathbb{R}^n$  satisfaz  $\langle \nabla f(x), d \rangle < 0$ , tem-se que  $d \in \mathcal{D}_f(x)$ .

Onde  $\langle \cdot, \cdot \rangle$  denota o produto interno (escalar) em  $\mathbb{R}^n$ .

*Demonstração.* Ver em Izmailov e Solodov (2005). □

Lembrando da definição de ângulo entre vetores da geometria analítica, tem-se que a desigualdade  $\langle \nabla f(x), d \rangle < 0$  é equivalente a dizer que a direção  $d$  forma um ângulo maior de  $90^\circ$  com a direção do gradiente  $\nabla f(x)$ . A Figura 4.4 que se segue, ilustra o conjunto de descida em um ponto, a partir das curvas de nível de uma função genérica.

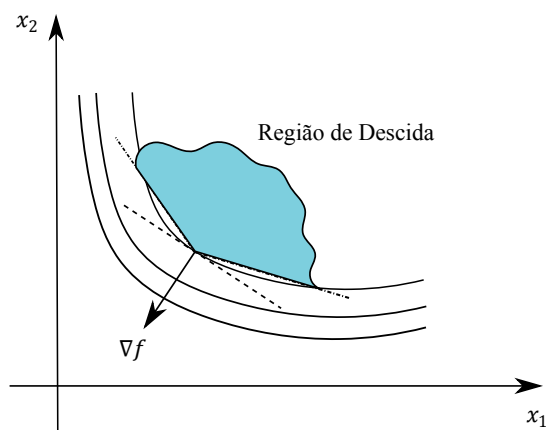


Figura 4.4: Conjunto de Descida  $\mathcal{D}_f(x)$ .

**Observação** - Segue que  $d = -\nabla f(x)$  é uma direção de descida de  $f$  no ponto  $x$ , mais ainda, é possível mostrar que  $d = -\nabla f(x)$  é a direção de máxima descida da função  $f$  no ponto  $x$  (IZMAILOV;SOLODOV, 2005; BRANDÃO, 2010).

Neste momento já está bem caracterizada a região de busca por direções de descida. Resta agora saber como calcular o tamanho do passo na direção de descida. Na definição de descida é garantido apenas que, para pequenos passos na direção de descida, a função diminuiria seu valor. O comprimento do passo é calculado a partir da observação do comportamento da função  $f$  ao longo da semi-reta a partir de  $x$  na direção de  $d$ . Por

este motivo, os procedimentos para o cálculo do tamanho do passo são chamados de *busca linear*. A seguir estão listadas as principais estratégias descritas na literatura para a busca linear para o método da máxima descida ou método do gradiente (IZMAILOV;SOLODOV, 2007; BRANDÃO, 2010).

- *Regra da Minimização Exata Unidimensional*

Como deseja-se diminuir o valor da função objetivo caminhando em sentido a uma direção de descida, nada é mais natural que definir, na  $k$ -ésima iteração, o tamanho do passo como sendo o argumento que satisfaz o seguinte problema de minimização unidimensional:

$$t_k = \underbrace{\arg \min}_{t>0} \{\varphi(t) = f(x_k - t\nabla f(x_k))\} \quad (4.30)$$

onde  $\varphi : (0, \infty) \rightarrow \mathbb{R}$  é uma função real de uma variável.

Primeiramente observe que, sendo  $-\nabla f(x_k)$  uma direção de descida, o problema 4.30, possui solução. Observe ainda que, caso a função  $f$  seja diferenciável, para toda direção de descida  $d_k$ , segue que

$$0 = \varphi'(t_k) = \langle \nabla f(x_k + t_k d_k), d_k \rangle = \langle \nabla f(x_{k+1}), d_k \rangle \quad (4.31)$$

Se  $\nabla f(x_{k+1}) \neq 0$ , do ponto do vista da geometria do problema, o ponto  $x_{k+1}$  é a inteseção da semi-reta, que inicia-se em  $x_k$  e com direção  $d_k$  com a curva de nível que passa pelo ponto  $x_{k+1}$ . Em particular, o método do gradiente move-se sempre em passos perpendiculares, isto é, duas direções de descida consecutivas são perpendiculares. De

fato, tem-se que

$$\begin{aligned} \langle x_{k+1} - x_k, x_{k+2} - x_{k+1} \rangle &= \langle -t_k \nabla f(x_k), -t_{k+1} \nabla f(x_{k+1}) \rangle \\ &= t_{k+1} t_k \langle \nabla f(x_k), \nabla f(x_{k+1}) \rangle. \end{aligned} \quad (4.32)$$

Note que, pela Equação 4.31, tem-se que  $\langle \nabla f(x_k), \nabla f(x_{k+1}) \rangle = 0$ , assim pode-se ver que dois passos consecutivos no método do gradiente são perpendiculares. A Figura 4.5 ilustra alguns passos consecutivos para o método do gradiente com a escolha do tamanho do passo através da minimização exata.

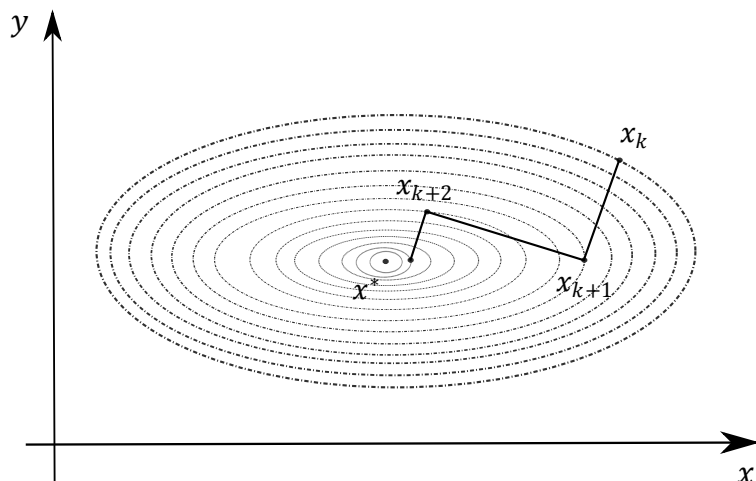


Figura 4.5: Geometria do método do Gradiente com passo obtido pela minimização exata.

Neste momento é conveniente ressaltar que para resolver, de forma aproximada, o problema 4.30 (minimização unidimensional) pode-se utilizar qualquer um dos métodos descritos na seção anterior deste capítulo.

A seguir é apresentado um teorema que garante a convergência do método do Gradiente, com a escolha do tamanho do passo pela minimização exata.

**Teorema 4.9.** *Seja  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  uma função com derivada contínua. Suponha que para um*

determinado  $x_0$  o conjunto de nível

$$L_f(f(x_0)) = \{x \in \mathbb{R}^n; f(x) \leq f(x_0)\} \quad (4.33)$$

seja fechado. Se  $\{x_k\}$  é a sequência gerada pelo Método do Gradiente, com ponto inicial  $x_0$ , e com o tamanho do passo determinado pela minimização exata, então tem-se que:

- i) a sequência  $\{x_k\}$  é limitada;
- ii) todo ponto de acumulação de  $\{x_k\}$  é ponto crítico da função  $f$ .

*Demonstração.* Ver em Izmailov e Solodov (2007). □

Para finalizar o assunto da minimização exata unidimensional, note que este processo pode ser demasiadamente caro do ponto de vista computacional. Sendo assim faz-se necessário utilizar alguma estratégia para encontrar  $t_k$  que minimize a função  $f$  na direção de descida  $-\nabla f(x_k)$ . Os métodos da bissecção, seção áurea, interpolação polinomial são bastante utilizados no intuito de encontrar uma solução aproximada para o problema de minimização unidimensional.

- *Regra Regularização Proximal*

Na Regularização Proximal a escolha do tamanho do passo, na  $k$ -ésima iteração, é o argumento que satisfaz o seguinte problema de minimização unidimensional:

$$t_k = \underbrace{\arg \min}_{t>0} \{\varphi(t) = f(x_k - t\nabla f(x_k)) + \alpha_k \|\nabla f(x_k)\|^2 t^2\} \quad (4.34)$$

onde  $\varphi : (0, \infty) \rightarrow \mathbb{R}$  é uma função real de uma variável e  $\alpha_k > 0$ .

- *Regra de Armijo*

Outra estratégia para escolha do passo no método do Gradiente é calcular o comprimento deste passo que resulta em um decréscimo suficiente da função  $f$  em relação ao valor  $f(x_k)$ . Esta idéia surge com a intenção do cálculo do tamanho do passo ser mais econômico do que na minimização exata unidimensional.

A Regra para escolha do tamanho do passo no Método do Gradiente denominada Regra de Armijo pode ser definida pelas condições abaixo. Suponha que a função  $f$  possua derivada primeira contínua e escolha parâmetros  $\bar{t} > 0$  e  $\sigma, \theta \in (0, 1)$ . Tome  $t_k = \bar{t}$ , então:

a) Verificar se a desigualdade

$$f(x_k - t_k \nabla f(x_k)) \leq f(x_k) + \sigma t_k \|\nabla f(x_k)\|^2 \quad (4.35)$$

é satisfeita ou não;

b) Se a Equação 4.35 não se satisfaz, então tome  $t_k = \theta \bar{t}$  e retorne ao passo a). Caso contrário, aceita-se  $t_k = \bar{t}$  como valor do comprimento de passo.

Izmailov e Solodov (2007) mostram que a Regra de Armijo está bem definida e até estabelecem uma cota inferior para o valor do comprimento de passo dado pela Regra de Armijo.

- *Regra do Gradiente Lipschitz*

A seguir será apresentado um conceito para auxiliar na definição da regra do Gradiente Lipschitz.

**Definição 4.10.** Diz-se que uma função  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  é Lipschitz contínua se existe uma cons-

tante  $L$  satisfazendo

$$\|f(x) - f(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathbb{R}^n. \quad (4.36)$$

O ínfimo das constantes  $L$  para os quais a desigualdade acima é válida é chamado de *Constante de Lipschitz*.

Na Regra do Gradiente Lipschitz, como o próprio nome diz, supõe-se que o gradiente da função  $f$  satisfaz a definição 4.10 com constante de Lipschitz  $L$ . A regra do Gradiente Lipschitz, para a escolha do tamanho do passo no método do Gradiente, é dada pela determinação de constantes  $\delta_1, \delta_2$  e  $t_k$  tais que

$$\frac{1}{2\delta_1} + \delta_2 < 1 \quad \text{e} \quad 0 < \delta_1 < t_k < \frac{2}{L(1 - \delta_2)}. \quad (4.37)$$

A seguir é apresentado um teorema sobre a convergência do Método do Gradiente, com o tamanho do passo determinado pela regra do Gradiente Lipschitz.

**Teorema 4.11.** *Suponha que uma função  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  possua derivada contínua e que o gradiente seja Lipschitz com constante  $L$ . Se  $\{x_k\}$  é a sequência gerada pelo Método do gradiente, com a escolha do tamanho do passo determinado pela Regra do Gradiente Lipschitz, então*

- i) existe  $\beta > 0$  tal que  $f(x_{k+1}) < f(x_k) - \beta\|x_{k+1} - x_k\|^2$ ;*
- ii)  $\{f(x_k)\}$  é monótona não crescente e convergente;*
- iii)  $\sum_{k=0}^{\infty} \|x_{k+1} - x_k\|^2 < \infty$ ;*
- iv)  $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$ ;*
- v) se  $\{x_k\}$  converge para  $\bar{x}$ , então  $\bar{x}$  é ponto crítico de  $f$ , i.e.,  $\nabla f(\bar{x}) = 0$ .*

---

*Demonstração.* Ver em Izmailov e Solodov (2007). □

Os autores Izmailov e Solodov (2007) apresentam, sob hipóteses variadas, algumas demonstrações de convergência do Método do Gradiente com vários tipos de escolha do tamanho do passo. Brandão (2010) também realiza um estudo sobre a convergência do método do Gradiente e traz alguns exemplos aplicações práticas do método.

## 4.5 Método de Newton e Métodos Quase-Newton

O método, a ser descrito nesta seção, foi proposto inicialmente por Isaac Newton em 1669 para encontrar raízes de funções polinomiais. Pouco tempo depois, em 1690, J. Raphson estendeu o método para funções de uma variável. Por este motivo é muito comum, na literatura, o método ser chamado método de Newton-Raphson. A consolidação do método está ligada a nomes famosos da matemática, tais como J. Fourier, L. A. Cauchy entre outros. Em 1818, Fourier provou que o método convergia quadraticamente desde que o ponto inicial fosse tomado em uma vizinhança da solução procurada, enquanto Cauchy (1829-1847) mostrou que o método se estende naturalmente para funções de várias variáveis e usou o método para provar a existência de raízes de algumas equações. Em 1916, os matemáticos Fine e Bennet deram mais algumas contribuições para o método. Fine (1916) provou a convergência para o caso  $n$ -dimensional sem a hipótese de existência de soluções. Bennet (1916) estendeu o resultado para o caso de dimensão infinita. Mais recentemente, em 1948, Kantorovich (1948) provou a existência de solução e a convergência do método para operadores  $\mathcal{T} : \mathcal{B}_1 \rightarrow \mathcal{B}_2$ , onde  $\mathcal{B}_1$  e  $\mathcal{B}_2$  são espaços de Banach e  $\mathcal{T}$  é um operador derivável qualquer. Os resultados de Bennet e Kantorovich merecem um destaque especial, pois foram obtidos antes da descoberta dos fundamentos da Análise Funcional; portanto, estes autores podem ser considerados como alguns dos precursores dessa área tão impor-

---

tante e bela da matemática. Mais informações sobre o desenvolvimento do método de Newton podem ser encontradas em Ypma (1995) e suas referências.

Sabe-se que tratar questões sobre a resolução de equações, ou mesmo mostrar que uma dada equação possui solução dentro de um certo espaço considerado, é objeto de interesse em diversas áreas da Ciência Pura e Engenharia. O método de Newton é uma poderosa ferramenta na obtenção de soluções de equações dos mais variados tipos. A grande importância desse método reside no fato de que sob algumas hipóteses é garantida a convergência, a uma taxa relativamente alta, para uma solução. O método de Newton também é usado para mostrar outros teoremas importantes na matemática. Por exemplo: teoremas de existência e unicidade de soluções para certas equações diferenciais veja por exemplo (KANTOROVICH;AKILOV, 1964; MOSER,1961), o teorema da função inversa e implícita (KRANTZ;PARKS, 2002) e o teorema do mergulho isométrico veja (NASH, 1956).

Como foi descrito, o método de Newton foi concebido inicialmente como uma ferramenta de resolução de sistemas de equações não lineares e não como um método de minimização. O método do Gradiente, descrito na seção anterior, é um método de minimização, porém o método de Newton, mesmo aplicado a um problema de otimização, i.e., à equação  $f'(x) = 0$ , tem a mesma “preferência” tanto para mínimos quanto para máximos, ou seja para quaisquer pontos estacionários.

O método de Newton é um método que, sob hipóteses em relação à segunda derivada de  $f$ , possui convergência quadrática. Assim uma iteração do método de Newton é significativamente mais cara do que uma iteração do método do Gradiente, que usa informações apenas de primeira ordem. Em contrapartida o método de Newton é usado como base de desenvolvimento dos métodos quase-newton (BRANDÃO, 2010; IZMAILOV;SOLODOV, 2007). Os métodos quase-Newton possuem iterações bem mais baratas do que as do



método de Newton e nem tanto mais caras do que as do método do Gradiente. Isto os tem tornado altamente atrativos para uso computacional e sem dúvidas estão entre os métodos determinísticos mais eficientes entre todos os algoritmos de otimização (SOLODOV; IZMAILOV, 2007).

### 4.5.1 O Método de Newton

O método de Newton foi originalmente concebido para resolver sistemas de equações do tipo

$$\varphi(x) = 0 \tag{4.38}$$

onde  $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}^n$  é uma função diferenciável. Seja  $x_k \in \mathbb{R}^n$  uma aproximação a uma solução  $x^*$  da Eq.4.38. Em torno de  $x_k$  podemos aproximar a Eq. 4.38 pela sua expansão de Taylor de primeira order, ou seja, a Eq. 4.38 pode ser reescrita da seguinte forma

$$\varphi(x_k) + \varphi'(x_k)(x - x_k) = 0 \iff x = x_k - J^{-1}(x_k)\varphi(x_k) \tag{4.39}$$

Onde  $J : \mathbb{R}^n \rightarrow \mathbb{R}^{n^2}$  é a matriz jacobiana de  $\varphi$ . A relação 4.39 é denominada de iteração do Método de Newton.

A Figura 4.6 a seguir, ilustra a aplicação do método de Newton para resolver uma equação em um caso unidimensional.

Observe que na iteração do Método de Newton, dada em 4.39, faz-se necessário que a matriz  $J(x_k)$  seja não singular, para que a iteração 4.39 esteja bem definida.

O método de Newton pode ser utilizado como método de otimização irrestrito, pois para resolver o problema de otimização de uma função  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  tem-se que encon-

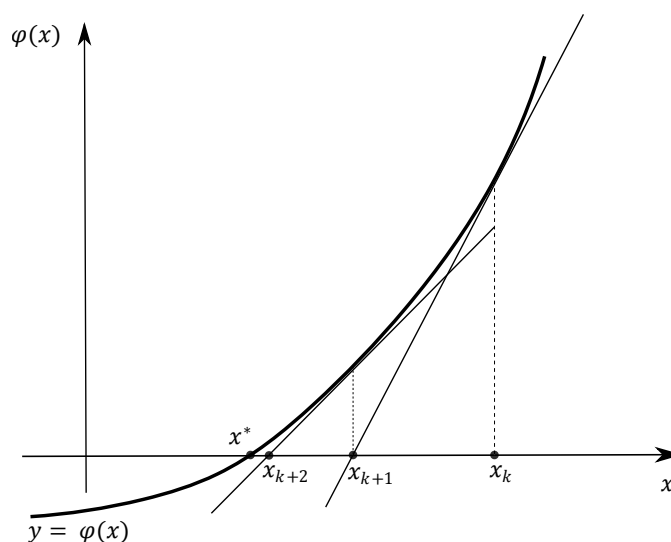


Figura 4.6: Geometria de aplicação do método de Newton.

trar pontos estacionários, isto é, encontrar soluções do sistema de equações não lineares  $\nabla f(x) = 0$ . Para escrever a iteração do método de Newton para encontrar soluções de  $\nabla f(x) = 0$  será preciso definir a derivada do gradiente da função  $f$ , isto é, a segunda derivada da função  $f$ , que é chamada de Matriz Hessiana de  $f$ , no caso desta ser uma função de várias variáveis.

**Definição 4.12.** *Seja  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  uma função que possui segunda derivada contínua. A matriz hessiana de  $f$ , é a matriz simétrica (Teorema de Schwarz (LIMA,2012)) definida por*

$$\mathcal{H}_f = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{pmatrix}.$$

Para a utilização do método de Newton em otimização, o objetivo então passa a ser encontrar pontos estacionários de  $f$ , isto é, resolver o sistema  $\nabla f(x) = 0$ . De forma

inteiramente análoga a realizada para obter a iteração do método de Newton, dada pela Eq.4.39, a iteração do método de Newton para encontrar soluções do sistema não linear  $\nabla f(x) = 0$  pode ser escrita da seguinte forma:

$$x_{k+1} = x_k - [\mathcal{H}_f(x_k)]^{-1} \nabla f(x_k) \quad (4.40)$$

Note que é necessário garantir que a matriz hessiana da função objetivo  $f$  seja não singular nos pontos da iteração de Newton. A seguir será apresentado um teorema que garante a boa definição da iteração do método de Newton bem como a convergência da sequência gerada pelo método.

**Teorema 4.13.** *Sejam  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  uma função com segunda derivada contínua e  $x^* \in \mathbb{R}^n$  tal que  $\nabla f(x^*) = 0$ . Suponha que a matriz hessiana de  $f$  seja não singular em  $x^*$  e que seja Lipschitz contínua com constante  $L$ . Nestas condições:*

- i) para qualquer ponto inicial  $x_0$ , suficientemente próximo de  $x^*$ , a iteração 4.40 gera uma sequência  $\{x_k\}$  bem definida e convergente para  $x^*$ ;*
- ii) a taxa de convergência é quadrática.*

*Demonstração.* Ver em Izmailov e Solodov (2007). □

Uma desvantagem do método de Newton é o fato da convergência ser apenas local, isto é, se o ponto inicial for tomado numa vizinhança de uma solução e sob certas hipóteses a sequência de Newton converge para a solução em questão. Uma das principais vantagens do método de Newton é o fato de que, sob certas condições, a taxa de convergência é quadrática. Porém o preço pago para isto é muito caro, pois necessita de hipóteses sobre a segunda derivada da função objetivo  $f$  e do ponto de vista computacional os cálculos envolvendo a segunda derivada podem ser demasiadamente demorados, em certas

aplicações da engenharia, tornando o método inviável. Uma saída bastante utilizada para contornar este problema é a utilização de metodologias que visam obter uma aproximação da matriz hessiana ao invés de calculá-la. Estas metodologias serão vistas na próxima seção deste capítulo.

### 4.5.2 Métodos Quase-Newton

Segundo Izmailov e Solodov (2007) os métodos Quase-Newton são bastante atrativos para uso prático computacional e são, em geral, os mais eficientes e mais utilizados entre todos os métodos determinísticos de otimização irrestrita. Considere o seguinte esquema iterativo:

$$x_{k+1} = x_k - Q_k \nabla f(x_k), \quad k = 0, 1, 2, \dots \quad (4.41)$$

onde, para todo  $k \in \mathbb{N}$ ,  $Q_k$  é uma matriz de ordem  $n$  simétrica positiva definida. Primeiramente note que a iteração dada pela Eq.4.41 é um método de descida, basta observar que se  $x_k$  não é um ponto estacionário de  $f$ , então

$$\langle \nabla f(x_k), -Q_k \nabla f(x_k) \rangle = -\langle \nabla f(x_k), Q_k \nabla f(x_k) \rangle < 0. \quad (4.42)$$

e portanto, pelo Lema 4.8,  $-Q_k \nabla f(x_k)$  é uma direção de descida. Observe ainda que se  $Q_k = I_n$  (matriz identidade de ordem  $n$ ), então o processo iterativo 4.41 é o método do gradiente com  $\alpha_k = 1$  e quando  $Q_k = [\mathcal{H}_f]^{-1}$ , a iteração 4.41 é o método de Newton Puro. Os métodos Quase-Newton possuem a forma 4.41, onde as matrizes  $Q_k$  são aproximações (num certo sentido) da matriz  $[\mathcal{H}_f]^{-1}(x^*)$ , para alguma solução  $x^*$ .

Há um teorema denominado de Teorema de Dennis-Moré que afirma que na otimização

de funções com segunda derivada contínua, um método, para ser eficiente, deve utilizar alguma aproximação da segunda derivada da função objetivo. Métodos que usam apenas aproximações de primeira ordem possuem convergência no máximo linear (IZMAILOV;SOLODOV, 2007).

Segundo Vanderplaats (1999), Izmailov e Solodov (2007) e Brandão (2010) o primeiro método quase-Newton é o método *Davidon – Fletcher – Powell* (DFP), que utiliza uma matriz  $Q_0$  positiva definida arbitrária e para todo  $k \in \mathbb{N}$  define a matriz  $Q_{k+1}$  da seguinte forma:

$$Q_{k+1} = Q_k + \frac{\xi_k(\xi_k)^t}{\langle \xi_k, \Delta f' \rangle} - \frac{(Q_k \Delta f')(Q_k \Delta f')^t}{\langle Q_k \Delta f', \Delta f' \rangle} \quad (4.43)$$

Note que todas as matrizes geradas através da Eq. 4.43 anterior são matrizes simétricas, positivas definidas e aproximam a matriz hessiana (IZMAILOV;SOLODOV,2007).

Segundo Izmailov e Solodov (2007) entre todos os métodos quase-Newton, o método *Broyden – Fletcher – Goldfarb – Shanno* (BFGS) é considerado o mais eficiente. No método BFGS a matriz  $Q_{k+1}$  é gerada através da seguinte forma:

$$Q_{k+1} = Q_k + \frac{(\xi_k - Q_k \Delta f')(\xi_k)^t + \xi_k(\xi_k - Q_k \Delta f')^t}{\langle \xi_k, \Delta f' \rangle} - \frac{\langle \xi_k - Q_k \Delta f', \Delta f' \rangle \xi_k(\xi_k)^t}{\langle \xi_k, \Delta f' \rangle^2} \quad (4.44)$$

Como no método DFP, pode-se observar que as matrizes geradas através do processo 4.44 são simétricas e positivas definidas.

## 4.6 Tratamento de Restrições

Nesta seção serão descritos, de forma sucinta, alguns métodos de tratamento de restrições. Para esta finalidade, o problema geral de otimização, a ser considerado, é formulado por:

$$\text{minimizar } f(x), \quad x \in \mathbb{R}^n \quad (4.45)$$

$$\begin{cases} h_l(x) = 0, & l = 1, \dots, L \\ g_j(x) \leq 0, & j = 1, \dots, J \\ x_i^{inf} \leq x_i \leq x_i^{sup}, & i = 1, \dots, n. \end{cases} \quad (4.46)$$

onde  $h_l$ ,  $l = 1, \dots, L$  e  $g_j$ ,  $j = 1, \dots, J$  são as restrições de igualdade e de desigualdade, respectivamente. Segundo Lobato (2008) as restrições em problemas de otimização são advindas de limitações operacionais, econômicas, físicas, ambientais, entre outras.

Devido tanto a sua simplicidade conceitual quanto a sua ampla utilização, serão realizadas apenas as descrições dos métodos de tratamento de restrições baseados em *Funções de Penalidade*. Nos métodos baseados em penalidade, o problema original, dado pelas Eq. 4.45 e 4.46, é aproximado por um problema irrestrito. Esta aproximação consiste em uma redefinição da função objetivo, de forma que violações de qualquer uma das restrições são penalizadas. A redefinição do problema original é expressa matematicamente por (VANDERPLAATS, 1999):

$$\varphi(x, r_p) = f(x) + r_p P(x) \quad (4.47)$$

onde  $\varphi(x, r_p)$  é a função Pseudo-Objetivo,  $r_p$  é chamado de fator de penalidade e  $P(x)$  é chamada de função de penalidade.

A classificação dos métodos de penalização dependem da Função de Penalização escolhida e possuem da seguinte forma (VANDERPLAATS, 1999):

- ***Penalização Externa***

Na Penalização Externa, a função Pseudo-Objetivo é penalizada apenas quando ocorre a violação de alguma das restrições, isto é, externamente ao conjunto viável. Neste caso, a Função de Penalização é dada por:

$$P(x) = \sum_{l=1}^L (h_l(x))^2 + \sum_{j=1}^J (\max [g_j(x), 0])^2 \quad (4.48)$$

Pode-se observar que a função pseudo-objetivo, na penalização externa, coincide com a função objetivo dentro do conjunto viável. Segundo Vanderplaats (1999) o método da Penalidade Externa deve ser iniciado com pequenos valores para o fator de penalidade, como por exemplo  $r_p = 1$ , e então ser corrigido a cada iteração por um fator multiplicativo, como por exemplo algum valor entre 2 e 10. Um ponto interessante a se destacar é que durante o processo iterativo na Penalização Externa o algoritmo pode convergir para a solução do problema original mesmo por pontos não viáveis.

#### • *Penalização Interna*

A Penalização Interna, também conhecida como Método das Barreiras, caracteriza-se por penalizar a função objetivo ainda no interior do conjunto viável. A Função de Penalidade deve ser definida de forma que a função objetivo seja penalizada quando os pontos gerados pelo processo iterativo se aproximam internamente da fronteira do conjunto viável. Neste caso a Função de Penalidade é definida por:

$$P(x) = \sum_{l=1}^L (h_l(x))^2 + \sum_{j=1}^J \frac{-1}{g_j(x)} \quad (4.49)$$

ou alternativamente por

$$P(x) = \sum_{l=1}^L (h_l(x))^2 + \sum_{j=1}^J \log(-g_j(x)) \quad (4.50)$$

Observe que o fator  $-1/g_j(x) > 0$  e tenderá ao infinito quando os pontos se aproximarem da fronteira do conjunto viável, isto é, na ativação da restrição  $g_j(x)$ . O mesmo ocorre ao fator  $\log(-g_j(x))$ . Em ambos os casos de definição da Função de Penalidade, o método gera uma barreira evitando a violação das restrições (IZMAILOV;SOLODOV, 2007).

Vanderplaats (1999) recomenda que o método seja iniciado com um fator de penalidade de valor elevado, como por exemplo  $10^6$  e que o mesmo seja atualizado por um fator multiplicativo compreendido entre 0 e 1.

No capítulo 7 será apresentado outro procedimento para tratamento de restrições. Porém, neste novo método, o fator de penalidade é mantido constante durante todo o processo iterativo.



# Capítulo 5

## Métodos Heurísticos de Otimização

Neste capítulo serão descritas algumas das principais heurísticas utilizadas na otimização de funções não lineares. São apresentados os principais mecanismos, denominados de operadores, contidos em cada heurística e que atuam “melhorando” as soluções de uma geração para outra com o objetivo de encontrar o conjunto solução em um espaço de busca viável definido pelas restrições. Métodos heurísticos de otimização têm sido utilizados com sucesso por diversos pesquisadores para resolverem problemas importantes da engenharia tais como: estimação de parâmetros cinéticos e dinâmicos da fermentação de etanol (WANG et al., 2001), identificação de forças externas em sistemas mecânicos (ROJAS, 2007), estimação de parâmetros de secagem em secadores rotativos (LOBATO et al., 2008), técnicas de modelagem e otimização aplicadas a problemas de projeto e de identificação de parâmetros (VIANA, 2008), otimização multiobjetivo para o projeto de sistemas (LOBATO, 2008), problemas inversos em transferência radiativa (NETO; BECCENERI, 2009), projeto ótimo robusto de absorvedores dinâmicos de vibrações (BORGES, et al., 2010), projeto de sistemas mecânicos não lineares (BORGES et al., 2013).

## 5.1 Algoritmos Genéticos - AG

Os Algoritmos Genéticos têm a sua base científica fundamentada na Teoria da Evolução de Darwin, onde indivíduos mais bem adaptados ao meio ambiente possuem maior probabilidade de sobrevivência e de deixarem descendentes para as próximas gerações. Os AG têm a sua origem principalmente no trabalho de John Henry Holland (1975) na Universidade de Michigan. Porém, foi um de seus alunos, David Goldberg, quem popularizou o método ao resolver um importante problema sobre transmissão de gás em uma tubulação em sua dissertação de mestrado (GOLDBERG, 1989). Ainda segundo Goldberg, os AG diferenciam-se dos algoritmos clássicos de otimização em basicamente três aspectos (GOLDBERG, 1989):

- exploram melhor o espaço de projetos por trabalharem com toda a população e não somente com um indivíduo;
- utilizam apenas o valor da função objetivo e não utilizam técnicas próprias de otimização determinística tais como as derivadas;
- não necessitam de nenhum conhecimento detalhado do problema, apenas de uma forma de avaliar o resultado.

Em analogia ao processo de evolução das espécies, as soluções candidatas são chamadas de indivíduos, também referidos na literatura como cromossomos, cadeias de caracteres, ou cadeias binárias (HAUPT e HAUPT, 1998; DEB, 2001). Nos AG uma população é criada e é então permitido a ela se adaptar ao espaço de projeto. Os indivíduos desta população são submetidos a cruzamentos entre eles, sendo que indivíduos mais *adaptáveis* possuem maior probabilidade de serem utilizados nestes cruzamentos. A medida que indica a adaptação de um indivíduo da população corresponde ao valor da função objetivo do

problema de otimização. Objetivando aumentar a diversidade da população é permitido aos indivíduos sofrerem algum processo de mutação. À medida que as gerações avançam melhores indivíduos são gerados e então tem-se que a avaliação da população descendente é sempre melhor que a avaliação de alguma população em uma geração ascendente. O processo geral dos AG podem ser resumidos através dos passos:

- gera-se, de forma aleatória, uma população inicial de indivíduos;
- realiza-se a codificação da população;
- efetua-se uma sequência repetitiva de procedimentos como: avaliar, selecionar, re-combinar (cruzamento - *crossover*), e modificar (mutação), gerando assim novas gerações com melhor avaliação;
- realizar o procedimento anterior até que algum critério de parada seja satisfeito.

Como os AG são baseados em um processo elitista de seleção dos indivíduos para reprodução, todos os procedimentos acima são efetuados com base na aptidão (melhor valor da função objetivo) dos indivíduos da população. Assim os indivíduos mais aptos terão uma maior probabilidade de serem selecionados para reprodução (cruzamento), passando para seus descendentes parte do seu material genético. A sequência de procedimentos é realizada até que um critério de parada seja satisfeito, sendo que um número máximo de gerações deve ser estabelecido a priori.

Quanto à codificação dos indivíduos da população, ela pode ser expressa na forma binária ou contínua (números reais ou pontos flutuantes). Nesta seção será descrito um algoritmo genético binário. Os AG usam um vocabulário emprestado da genética. Abaixo são apresentados os termos mais comuns que serão utilizados na descrição do algoritmo genético:

- **cromossomo** - É a estrutura que representa uma variável de projeto, denominado como sendo um indivíduo. Cada cromossomo deve possuir  $n$  subestruturas que correspondem às  $n$  coordenadas de cada variável de projeto no espaço de busca;
- **gene** - É a unidade básica de hereditariedade, sendo carregada em um cromossomo na forma de DNA. Cada cromossomo deve possuir  $n$  genes, isto é, cada gene representa uma variável de projeto  $x_i$ . Na codificação de parâmetros binária deve-se estabelecer qual é o comprimento de cada gene, isto é, qual a quantidade de bits que representará cada gene;
- **alelos** - São as unidades presentes em cada gene do cromossomo. Cada gene deve possuir  $m$  alelos que é a quantidade de bits de cada gene. O comprimento  $m$  depende da precisão requerida para o problema e da amplitude do intervalo definido pelas restrições laterais.

A Equação 5.1 abaixo exemplifica o que foi descrito acima sobre a estrutura de um cromossomo (indivíduo). Observe que, neste caso, cada gene foi descrito com cinco alelos.

$$\text{Cromossomo} = \left[ \begin{array}{cccc} x_1 & x_2 & x_i & x_n \\ \underbrace{11010}_{\text{gene 1}} & \underbrace{00101}_{\text{gene 2}} & \dots \underbrace{10101}_{\text{gene } i} & \dots \underbrace{10011}_{\text{gene } n} \end{array} \right] \quad (5.1)$$

A Figura 5.1 abaixo representa um fluxograma com o esquema de otimização por algoritmos genéticos.

Nas próximas subseções serão apresentados os operadores de seleção, cruzamento e mutação. Na prática tem-se verificado que a consistência de um AG é garantida por estes três operadores (HAUPT;HAUPT, 1998).

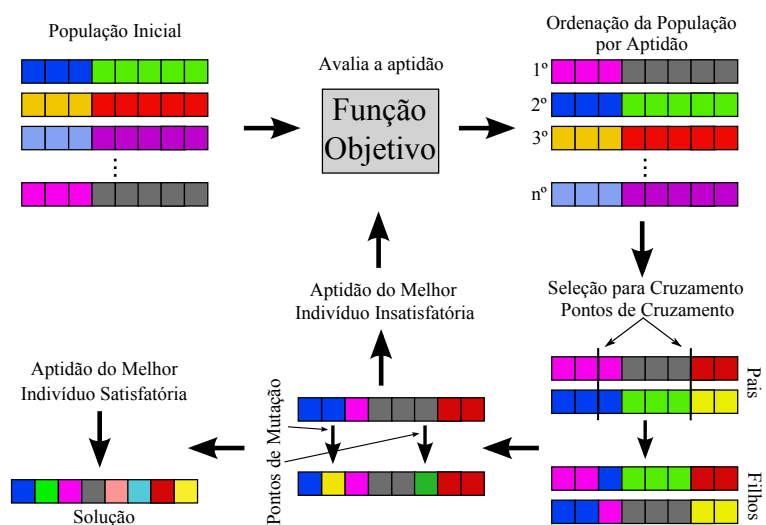


Figura 5.1: Fluxograma esquemático do processo de otimização de um AG.

### 5.1.1 Operador Seleção

Após a realização do processo de escolha da população inicial, que pode ser realizado de forma aleatória ou definida pelo usuário, e da codificação das variáveis, faz-se necessário a seleção dos indivíduos que serão usados nos cruzamentos. Dois pais selecionados para cruzamento geram apenas dois descendentes, porém a dimensão da população inicial permanece constante durante todo o processo de otimização. Variações destas idéias são encontradas em vários trabalhos. O processo de seleção dos AG é elitista, e portanto tendencioso no sentido de produzir membros mais adaptados e eliminar os menos adaptados. Desta forma os cruzamentos são realizados com os melhores indivíduos e espera-se que a convergência do algoritmo seja rápida. Abaixo são descritos três tipos de seleção.

- **Seleção Aleatória** - como o próprio nome já indica, a seleção é realizada de forma totalmente aleatória para um posterior cruzamento;
- **Seleção por Torneio** - dois indivíduos são escolhidos de forma aleatória. A seguir gera-se um número aleatório  $\rho \in [0, 1]$ . Se  $\rho$  for menor que um número previamente

definido, que determina o quanto o operador de seleção será elitista, então o indivíduo com maior aptidão é escolhido e o outro eliminado; caso contrário, o menos apto é escolhido e o mais apto é eliminado.

- **Seleção pela Roleta** - uma forma bastante utilizada para realizar a seleção é o Método da Roleta (*roulette wheel*), no qual a probabilidade de seleção de um indivíduo é diretamente proporcional ao valor correspondente da função objetivo, que é uma medida de sua adaptação. Se  $f_i$  indica a medida de adaptação do  $i$ -ésimo indivíduo ( $Ind_i = i$ -ésimo indivíduo) da população, pode-se associar este indivíduo a uma probabilidade  $p_i = f_i / \sum_{i=1}^{N_p} f_i$ , onde  $N_p$  é o tamanho da população. Em seguida a roleta, de área 1, é dividida em  $N_p$  partes, cada uma com área  $p_i$ ,  $i = 1, 2, \dots, N_p$  em ordem decrescente dos valores de  $p_i$ . Criam-se os seguintes  $N_p$  números

$$q_1 = p_1, q_2 = p_1 + p_2, \dots, q_i = p_1 + p_2 + \dots + p_i, \dots, q_{N_p} = p_1 + p_2 + \dots + p_{N_p}$$

onde  $p_1 \geq p_2 \geq \dots \geq p_{N_p}$  e portanto  $q_1 \leq q_2 \leq \dots \leq q_{N_p} = 1$  isto é, as probabilidades foram colocadas em ordem decrescente de tamanho. Geram-se  $N_p$  números aleatórios no intervalo  $[0, 1]$ . Para cada número aleatório gerado, seja  $q_j$  o primeiro valor da seqüência crescente  $q_1, q_2, \dots, q_{N_p} = 1$  tal que  $q_j$  seja maior que este número aleatório. Então o indivíduo  $Ind_j$  é selecionado. Repete-se este procedimento até que todos os números aleatórios sejam testados. Observa-se que este procedimento privilegia os indivíduos mais aptos, que podem ser selecionados mais de uma vez, ao passo que indivíduos menos aptos podem até não serem selecionados para reprodução. Depois de selecionados, os cromossomos (indivíduos) passam para a próxima etapa do Algoritmo Genético, que é denominada Cruzamento.

### 5.1.2 Operador Cruzamento

No processo de reprodução serão usados os indivíduos (cromossomos) determinados pelo processo de seleção. O operador de cruzamento é a primeira forma do algoritmo explorar o espaço das variáveis de projeto (espaço de busca) e evitar uma convergência prematura, isto é, evitar ótimos locais (HAUPT;HAUPT,1998). Existem diversas formas de realização do cruzamento. Será descrita aqui uma forma de cruzamento que envolve dois pais que irão produzir dois descendentes. Na literatura pode-se observar outros tipos de cruzamentos nos trabalhos de Goldberg (1989), Michalewicz (1996) e Reeves (2003).

Uma probabilidade de cruzamento  $P_c$ , definida a priori pelo usuário, é usada para determinar se o cruzamento será realizado. Pode-se utilizar como estratégia de cruzamento a de se escolher um ou dois pontos de cruzamento na cadeia binária, e efetuar a troca dos dígitos 0's e 1's entre os dois pais para todos os dígitos entre os pontos de cruzamento (para o caso de dois pontos) ou efetuar a troca de 0's e 1's entre os pais a partir deste ponto de cruzamento (para o caso de um ponto). A Figura 5.2 a seguir mostra um cruzamento a partir de um ponto.

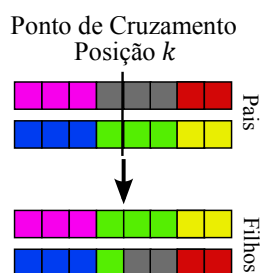


Figura 5.2: Representação de um operador de cruzamento pontual.

### 5.1.3 Operador Mutação

A mutação protege a busca genética de uma perda prematura de bom material genético. Uma estratégia simples de mutação é a mutação de ponto único, que muda o bit “0” para “1” e vice versa em algumas posições do cromossomo. Uma forma de realizar a mutação é selecionar aleatoriamente a posição em um cromossomo, obedecendo a uma probabilidade de mutação  $P_m$  pré-estabelecida e mudar o valor do bit. Outra estratégia utilizada com frequência é para cada bit gerar um número aleatório no intervalo  $[0,1]$  e comparar este número aleatório com a probabilidade de mutação  $P_m$ . Caso o número aleatório gerado seja menor que a probabilidade de mutação  $P_m$  é feita a mutação nos bits correspondentes. Geralmente usa-se  $P_m$  igual a 1% (HAUPT;HAUPT, 1998). Depois de realizadas as mutações, os cromossomos resultantes compõem a próxima geração. A Figura 5.3 a seguir, representa um processo de mutação em um cromossomo.

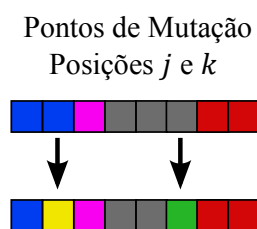


Figura 5.3: Representação de um operador de mutação em dois pontos.

De acordo com Filho et al. (1994), desde que Holland propôs seu modelo básico para um AG, tem sido introduzidos vários operadores genéticos na composição de um AG. Eles são, em geral, versões de cruzamento e alteração dos processos genéticos adaptados às necessidades de problemas específicos.



## 5.2 Evolução Diferencial

Com o objetivo de resolver o problema de ajuste polinomial de Chebychev, Kenneth Price e Ranier Storn desenvolveram o algoritmo de Evolução Diferencial (*Differential Evolution - DE*) (STORN;PRICE,1995). A principal motivação para a criação do método ED foi a lenta taxa de convergência e a dificuldade de determinação de alguns parâmetros apresentada pelo algoritmo denominado Recozimento Genético, que fora anteriormente proposto por Price em 1994. O algoritmo Recozimento Genético é uma técnica de otimização estocástica, baseada em população, que alia informações dos algoritmos Recozimento Simulado e Algoritmos Genéticos (NETO;BECCENERI, 2009). Outro aspecto interessante que motivou o desenvolvimento do algoritmo DE foi sua eficiência na resolução de problemas complexos utilizando conceitos relativamente simples. Uma grande vantagem do algoritmo de DE é a capacidade dele em trabalhar com funções não lineares até mesmo descontínuas, sem uso de derivadas.

Segundo Lobato (2008) a eficiência do algoritmo DE está associada às seguintes características:

- é uma heurística baseada em população;
- não requer o conhecimento de derivadas da função objetivo e funciona bem mesmo quando é aplicado em funções descontínuas e com multi-modalidade;
- trabalha com variáveis contínuas, discretas, inteiras e/ou binárias ao mesmo tempo sem ter que reestruturar o problema original;
- funciona bem mesmo quando a população inicial é pequena.

Storn e Price (1997) orientam a escolha dos fatores de controle do algoritmo DE.

---

Segundo os autores, o fator de perturbação  $F_p$ , a probabilidade de cruzamento  $P_c$  e o tamanho da população  $N_p$  devem satisfazer as condições:

- a população inicial deve ser gerada o mais próximo possível da superfície fronteira do domínio da função objetivo;
- a probabilidade de cruzamento  $P_c$  deve ser escolhida tal que  $P_c \in [0, 1]$ . Caso não ocorra a convergência ao usar valores pequenos para  $P_c$ , deve-se então utilizar  $P_c \in [0, 8; 1]$ ;
- o tamanho da população deve ser um número de 5 a 10 vezes o número de variáveis de projeto;
- o fator de perturbação  $F_p$  deve ser considerado no intervalo  $[0, 2; 2]$ ;
- quanto maior for o tamanho da população, menor deve ser o fator de perturbação  $F_p$ ;

A ideia central do algoritmo DE é gerar vetores, chamados de vetores modificados ou doadores, pela adição de uma diferença vetorial ponderada entre dois vetores aleatórios (ou não) da população a um terceiro vetor. Esta operação é chamada de **Mutação**. As coordenadas deste novo indivíduo (vetor) doador são misturadas com as coordenadas de outro vetor escolhido aleatoriamente (ou o melhor indivíduo da geração), denotado por vetor alvo ou vetor a ser substituído, resultando em outro vetor chamado de vetor experimental. Esta operação é chamada de **Cruzamento**. Se o valor da função objetivo do vetor experimental for menor que o valor da função objetivo do vetor alvo, então o vetor experimental será o vetor alvo da próxima geração. Esta operação é chamada de **Seleção**. O algoritmo é executado até algum critério de parada ser satisfeito. Nas próximas seções

serão descritos com mais detalhes cada um dos operadores citados acima e também serão realizados alguns comentários sobre o algoritmo DE.

### 5.2.1 Operador de Mutação

Considere  $x_\alpha, x_\beta$  e  $x_\gamma$  três vetores distintos, no espaço de projetos, escolhidos de forma aleatória a partir de uma população dada. Para gerar um vetor doador  $V$ , pode-se proceder da seguinte forma:

$$V = x_\alpha + F_p(x_\beta - x_\gamma) \quad (5.2)$$

Existem outras propostas de geração do vetor doador onde o vetor  $x_\alpha$  é o melhor indivíduo da geração corrente ( $x_{best}$ ) e/ou a diferença vetorial, que aparece acima com dois vetores, é realizada com quatro vetores, isto é, vetores doadores podem ser obtidos através das fórmulas:

$$V = x_{best} + F_p(x_\beta - x_\gamma) \quad (5.3)$$

ou

$$V = x_\alpha + F_p(x_\beta - x_\gamma + x_\mu - x_\eta) \quad (5.4)$$

ou ainda por

$$V = x_{best} + F_p(x_\beta - x_\gamma + x_\mu - x_\eta) \quad (5.5)$$

onde os índices  $\alpha, \beta, \gamma, \mu$  e  $\eta$  são distintos entre si e pertencem ao conjunto  $\{1, 2, \dots, N_p\}$ ,

onde  $N_p$  é o tamanho da população. A Figura 5.4 a seguir ilustra a geração de um vetor doador a partir de três vetores  $x_\alpha, x_\beta$  e  $x_\gamma$  distintos.

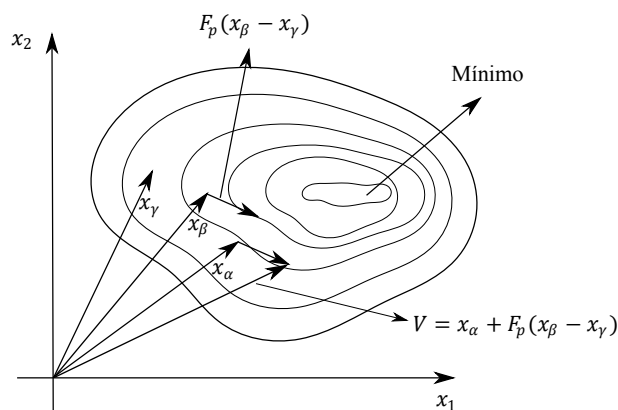


Figura 5.4: Geração de um vetor doador no algoritmo DE. Adaptada de (PRICE;STORN;LAMPINEN, 2005).

## 5.2.2 Operador de Cruzamento

Este operador tem como objetivo aumentar a diversidade dos vetores que sofrem a mutação, isto é, dos vetores doadores. Sejam  $x_\xi$  e  $V$  vetores alvo e doador respectivamente. Utilizando as coordenadas dos vetores  $x_\xi = (x_\xi^1, x_\xi^2, \dots, x_\xi^{N_{var}})$  e  $V = (v_1, v_2, \dots, v_{N_{var}})$ , cria-se um novo vetor  $U = (u_1, u_2, \dots, u_{N_{var}})$ , denominado vetor experimental, através da seguinte regra

$$u_i = \begin{cases} v_i, & \text{se } rand_i \leq P_c \\ x_\xi^i, & \text{se } rand_i > P_c, \end{cases} \quad (5.6)$$

onde  $rand_i$  é um número aleatório no intervalo  $[0, 1]$  e a probabilidade de cruzamento  $P_c$  é tal que  $P_c \in [0, 1]$ . Este tipo de cruzamento é denominado de cruzamento binomial, pois é obtido por experimentos entre dois vetores, alvo e doador. A Figura 5.5 a seguir ilustra um cruzamento binomial em um caso com sete variáveis de projeto.

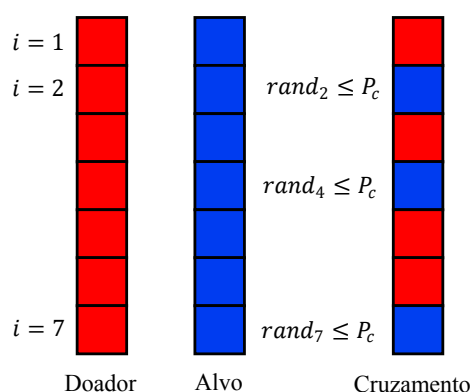


Figura 5.5: Cruzamento binomial no algoritmo DE. Adaptada de (PURCINA, 2010).

Outro tipo de cruzamento bastante utilizado na literatura é o cruzamento exponencial (STORN;PRICE, 1997). No cruzamento exponencial as coordenadas do vetor experimental são iguais às do vetor doador enquanto o número aleatório for menor que a probabilidade de cruzamento. A partir do momento que o número aleatório for maior que a probabilidade de cruzamento, as coordenadas do vetor experimental passam a ser obtidas do vetor alvo. Oliveira (2006) descreve com detalhes outros tipos de cruzamentos que podem ser realizados no algoritmo DE.

### 5.2.3 Operador de Seleção

O operador de seleção tem a função de selecionar os melhores indivíduos para a próxima geração. Assim, se o vetor experimental  $U$  possui menor valor quando avaliado na função objetivo do que o valor na função objetivo do vetor alvo  $x_\xi$  então o vetor experimental passa para a próxima geração, caso contrário ele é descartado. O operador de seleção pode ser descrito pela seguinte regra:

$$\text{Se } \begin{cases} f(U) \leq f(x_\xi) \implies x_\xi = U; \\ f(U) > f(x_\xi) \implies x_\xi = x_\xi. \end{cases} \quad (5.7)$$

### 5.2.4 Estratégias do Algoritmo DE

As variações das formas de realizar as operações de mutação e de cruzamento correspondem a estratégias distintas de execução do algoritmo DE. A notação utilizada para representar a estratégia escolhida é **DE / x / y / z**, onde:

*i)* **x** indica, na operação de mutação, o vetor que será somado à diferença ponderada.

Pode ser *rand* ou *best*;

*ii)* **y** indica o número de diferenças ponderadas usadas na soma com o vetor  $x$  na operação de mutação;

*iii)* **z** indica o tipo de cruzamento que será utilizado na execução do algoritmo. Pode ser binomial ou exponencial.

Não há como dizer qual é a melhor estratégia a ser escolhida quando se realiza uma simulação. Na literatura encontram-se trabalhos onde os autores mencionam que uma estratégia específica apresentou melhores resultados no problema estudado, porém também é possível encontrar autores que mencionaram que a escolha da estratégia pouco influenciou nos resultados obtidos. Por exemplo, Angira e Babu (2005) e Babu e Anbarasu (2005) observaram que a estratégia **DE /rand/1/bin** apresentou os melhores resultados em suas simulações, enquanto Oliveira (2006), em sua pesquisa de mestrado, realizou diversas simulações para diferentes casos e estratégias e concluiu que a escolha da estratégia não influenciou de forma significativa nos resultados e na ordem de convergência do algoritmo

ED. A Tabela 5.1 indica algumas das principais estratégias utilizadas no algoritmo de evolução diferencial.

Tabela 5.1: Principais estratégias para mutação e cruzamento no algoritmo DE

Estratégia	Operador Mutação	Notação
1	$V = x_\alpha + F_p(x_\beta - x_\gamma)$	DE/rand/1/bin ou exp
2	$V = x_\alpha + F_p(x_\beta - x_\gamma + x_\mu - x_\eta)$	DE/rand/2/bin ou exp
3	$V = x_{best} + F_p(x_\beta - x_\gamma)$	DE/best/1/bin ou exp
4	$V = x_{best} + F_p(x_\beta - x_\gamma + x_\mu - x_\eta)$	DE/best/2/bin ou exp
5	$V = x_{old} + F_p(x_{best} - x_{old} + x_\alpha - x_\beta)$	DE/rand-to-best/2/bin ou exp

onde  $x_{old}$  é o melhor vetor da geração anterior e  $x_{best}$  é o melhor vetor da geração corrente.

### 5.3 Colônia de Vagalumes

Os vagalumes são sem dúvida um dos insetos mais fascinantes da natureza. São insetos noturnos da família *Lampyridae* (ordem Coleoptera) e são famosos por sua característica bioluminescente. Habitam principalmente as regiões tropicais e temperadas e sua população é composta por mais de 1900 espécies (ENCYCLOPEDIA BRITANICA, 2009).

Baseado no comportamento destes insetos, Xin-She Yang propôs, na Universidade de Cambridge, um algoritmo heurístico de otimização (YANG, 2008). Segundo Yang a biologia ainda não possui um conhecimento completo de todas as utilidades que a luminescência pode trazer aos vagalumes, mas pelo menos três funções já foram identificadas:

- a luminescência é uma ferramenta de comunicação e de atração para potenciais parceiros de reprodução;
- serve como isca para a atração de alguma eventual presa;

- também serve como um mecanismo de defesa, pois é um sinal de alerta aos predadores lembrando-os que os vagalumes possuem um “sabor amargo”.

O Algoritmo de Colônia de Vagalumes (ACV) é baseado principalmente na primeira característica, isto é, na atração de parceiros para reprodução. Esta característica é vista em algumas espécies de vagalumes, onde a taxa de intermitência e a intensidade de cada *flash* é parte essencial do mecanismo de atração do sexo oposto para acasalamento. Na maioria dos casos, as fêmeas é que são atraídas pelo brilho emitido pelos machos. Observa-se também que quando há uma grande quantidade de vagalumes numa mesma região, então há uma sincronização na emissão dos *flashes*, que torna evidente uma característica de organização emergente (YANG, 2008).

### 5.3.1 O Algoritmo de Colônia de Vagalumes

Para a implementação do ACV, Yang (2008) definiu três regras simplificadoras para o delineamento da execução do algoritmo:

- qualquer vagalume da colônia poderá ser atraído ou atrair;
- a atratividade de um vagalume é diretamente proporcional ao brilho emitido e inversamente proporcional à distância entre os vagalumes (esta regra é baseada no comportamento real destes insetos);
- o brilho emitido é determinado em comparação com seu valor na função objetivo, isto é, quanto melhor avaliado maior será o seu brilho.

Para entender o ACV deve-se compreender como se dá a variação da intensidade do brilho percebido pelo vagalume e como é formulada a atratividade entre eles. Ainda segundo Yang (2008), a atratividade de um vagalume é determinada pela intensidade do



brilho emitido por ele, e esta intensidade é função de sua avaliação. A atratividade entre dois vagalumes é inversamente proporcional à distância entre eles. Sendo assim, deve ser escolhida uma função decrescente em relação à distância  $r = d(x_i, x_j)$  entre os vagalumes. Tem sido amplamente utilizada a seguinte função:

$$\beta = \beta_0 e^{-\gamma r^2} \quad (5.8)$$

onde  $\beta_0$  e  $\gamma$  são parâmetros pré-determinados do algoritmo: atratividade máxima (quando  $r = 0$  e geralmente  $\beta_0 = 1$ ) e coeficiente de absorção da luz, respectivamente.

Para explorar eficazmente o espaço de projetos, cada vagalume do enxame move-se iterativamente de acordo com dois fatores:

- a atratividade de outros membros do enxame com maior intensidade de luz emitida que varia de acordo com a distância;
- um vetor de passo aleatório, para evitar uma convergência prematura.

Assim, na  $k$ -ésima iteração, a movimentação do  $i$ -ésimo vagalume  $x_i^k$  em direção ao vagalume mais atrativo  $x_j^k$  (pelo menos pelo ponto de vista do vagalume  $x_i^k$ ) é definida pela equação:

$$x_i^{k+1} = x_i^k + \beta(x_j^k - x_i^k) + \alpha \left( rand - \frac{1}{2} \right) \quad (5.9)$$

em que, o segundo termo da equação insere o fator de atratividade  $\beta$  e o terceiro termo, regulado pelo parâmetro  $\alpha$ , permite a inserção de certa aleatoriedade no caminho percorrido pelo vagalume e  $rand$  é um número aleatório entre 0 e 1. Todos vagalumes da colônia se movimentam de acordo com a Eq. 5.9 e a movimentação dos vagalumes, em direção a solução ótima, é realizada até que algum critério de parada seja satisfeito. Observe que

um vagalume não necessariamente segue o vagalume mais brilhante (melhor avaliado na função objetivo) da colônia e sim aquele que é mais atrativo a ele. Isto faz com que haja uma maior diversidade na população de vagalumes, pois se todos seguissem o vagalume mais brilhante a população se estagnaria em poucas gerações.

## 5.4 Enxame de Partículas - EP

O algoritmo de Otimização por Enxame de Partículas (do inglês *Particle Swarm Optimization*) é uma heurística baseada em população e que foi proposta por James Kennedy e Russel Eberhart (KENNEDY;EBERHART,1995). Surgiu a partir de processos que modelam o comportamento social de algumas espécies de pássaros e cardumes de peixes onde cada ave (peixe) é uma partícula e o bando (cardume) é o enxame. Existem vários modelos de comportamento social, dentre eles, Kennedy e Eberhart se interessaram por um modelo particular desenvolvido por um biólogo chamado Frank Heppner.

Considere, por hipótese, que um grupo de pássaros está procurando por comida ou local para descanso, em certa região. Suponha que nesta região há apenas um local de comida ou descanso e que os pássaros não sabem, a priori, onde está este lugar. Qual é a melhor estratégia para procurar este lugar? A estratégia escolhida por muitos grupos de pássaros é a de seguir um pássaro que está mais próximo da comida ou do descanso. Através de regras simples que os pássaros usam para se movimentarem, um pássaro quando sai do bando para pousar na comida ou no local de descanso acaba atraindo os pássaros mais próximos, isto é, os indivíduos aprendem com o sucesso da "melhor" ave do bando. Outro fato a ser considerado é que cada indivíduo do bando faz uso de informações pessoais para encontrar um local para descanso ou uma fonte de alimento. Assim faz-se necessário um ajuste da capacidade individual de procurar uma boa solução (exploração -

*exploration*) e da capacidade de tirar proveito das informações do bando para alcançar seu próprio sucesso (proveito - *exploitation*). Quando há pouca exploração, a resposta pode convergir para a primeira boa solução encontrada e quando há pouco proveito, pode ser até que a resposta ótima nunca seja encontrada. Portanto, para aumentar as chances de alcance da resposta esperada, deve-se balancear estes dois comportamentos observados: o conhecimento individual e o de socialização dos indivíduos do enxame.

### 5.4.1 O Algoritmo EP

No enxame definido pelo algoritmo EP, as partículas que o compõe comunicam-se para compartilhar o conhecimento que cada uma adquiriu e, desta forma, aumentam sua capacidade de tomada de decisão, não baseada apenas em um conhecimento individual de cada partícula, mas no conhecimento do enxame. Abaixo são estabelecidas as notações necessárias para o entendimento do algoritmo. Cada partícula  $x_i$  do enxame, na iteração  $k$  será representada pelos seguintes vetores:

- a posição atual da partícula ( $n$ -dimensional):  $x_i^k = (x_{1i}^k, x_{2i}^k, \dots, x_{ni}^k)$ ;
- a melhor posição encontrada pela partícula até o momento:  $p_{best}^i$ ;
- a sua velocidade atual:  $v_i^k = (v_{1i}^k, v_{2i}^k, \dots, v_{ni}^k)$ .

O enxame de partículas também guarda um vetor  $p_{best}^{swarm}$ , que representa seu conhecimento social. Este vetor armazena a melhor posição encontrada, em todas as iterações, quando todas partículas do enxame são consideradas.

O EP faz uso de um vetor de velocidades e um vetor de posição para modelar o comportamento (movimento) das partículas do enxame. O vetor de velocidades e o vetor

de posição das partículas são atualizados pelas seguintes equações:

$$v_i^{k+1} = wv_i^k + c_1r_1\frac{p_{best}^i - x_i^k}{\Delta t} + c_2r_2\frac{p_{best}^{swarm} - x_i^k}{\Delta t} \quad (5.10)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1}\Delta t \quad (5.11)$$

onde  $r_1$  e  $r_2$  são números aleatórios uniformemente distribuídos entre 0 e 1 e  $i = 1, 2, \dots, N_p$ , sendo que  $N_p$  é o tamanho da população. A figura abaixo ilustra, para um caso bidimensional, a atualização da velocidade quando consideradas duas partículas.

Existem três parâmetros que devem ser controlados pelo usuário na simulação do algoritmo EP, são eles: a inércia  $w$  da partícula, e os dois parâmetros  $c_1$  e  $c_2$  (cognitivo e social respectivamente). A inércia controla a capacidade do algoritmo em explorar, isto é, um valor alto de  $w$  facilita um comportamento mais global, enquanto um valor baixo facilita um comportamento local (VENTER;SOBIESZCZANSKI-SOBIESKI, 2002). Os parâmetros de confiança  $c_1$  e  $c_2$  indicam o quanto uma partícula confia em si ( $c_1$ ) e no enxame ( $c_2$ ). A Figura 5.6 ilustra o movimento de uma partícula no algoritmo EP.

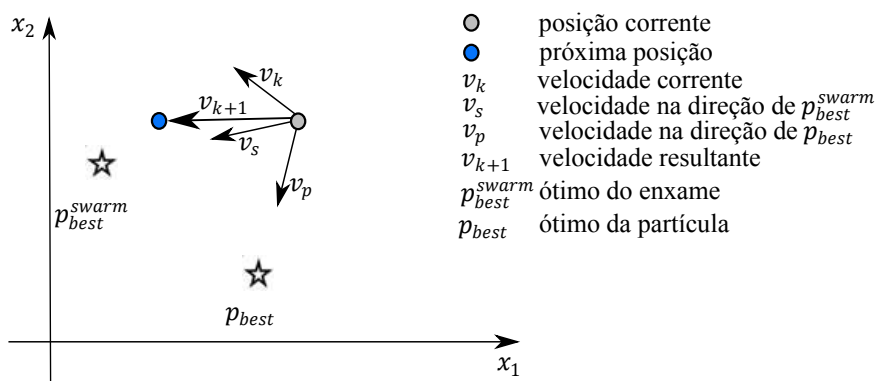


Figura 5.6: Movimento de uma partícula no algoritmo EP. Adaptada de (VIANA, 2008).

Após a definição dos parâmetros do algoritmo EP e da criação de uma população inicial, o processo de otimização pelo algoritmo de enxame de partículas consiste em uma

seqüência repetitiva das operações:

- atualizar o vetor velocidade;
- atualizar o vetor posição;
- avaliar a função objetivo em todas partículas do enxame;
- verificar se algum critério de parada é satisfeito.

## 5.5 Recozimento Simulado - RS

O algoritmo de otimização denominado Recozimento Simulado (do inglês *Simulated Annealing*) é baseado no processo físico do resfriamento de um metal em estado de fusão. Possui suas bases em ideias da mecânica estatística e no algoritmo de simulação proposto por Metropolis et al. (1953). O RS foi apresentado como um método de otimização por Kirkpatrick et al. (1983).

No processo do recozimento a temperatura de um metal em estado de fusão é reduzida bem lentamente, para garantir que baixos níveis de energia interna sejam alcançados e então obter uma estrutura cristalina pura do metal. Segundo Neto e Becceneri (2009) se não for feito assim os cristais resultantes terão muitos defeitos ou a substância pode se transformar em uma estrutura vítrea, que é uma estrutura apenas localmente ótima.

A função objetivo que se deseja minimizar corresponde à energia do sistema. A temperatura será apenas um parâmetro de controle. Visando a “fuga” de mínimos locais e conseqüentemente a convergência para um mínimo global, o algoritmo proposto por Metropolis aceita, no processo iterativo, passos que resultam no aumento da função objetivo. Em cada passo do algoritmo, a uma mesma temperatura, são dados deslocamentos aleatórios em cada átomo do sistema e calcula-se a variação da energia  $\Delta E = \Delta E_{i+1} - \Delta E_i$ .

Se  $\Delta E < 0$  o deslocamento é aceito e a nova configuração do sistema é tomada como o ponto de partida para a próxima iteração. Caso  $\Delta E > 0$ , a nova configuração pode até ser aceita. Gera-se um número aleatório  $\rho$ , com distribuição uniforme (também é comum a utilização da distribuição de Cauchy) no intervalo  $[0, 1]$ , e este número é comparado à probabilidade

$$P(\Delta E) = \exp\left(\frac{\Delta E}{k_B T}\right). \quad (5.12)$$

A nova configuração será aceita se  $\rho < P(\Delta E)$  caso contrário é rejeitada e a configuração anterior é tomada novamente como ponto de partida para o algoritmo. Este critério que possibilita a aceitação de um ponto que piora o valor da função objetivo é chamado de critério de Metrópolis e é de fundamental importância para a fuga de mínimos locais. Na equação acima  $T$  é a temperatura e  $k_B$  é uma constante chamada constante de Boltzmann. A constante  $k_B$  é apenas um fator de escala e normalmente é tomada como sendo 1. Repetindo-se este procedimento diversas vezes, a uma temperatura  $T$ , simula-se o movimento dos átomos em um material. Em cada temperatura o algoritmo é executado um número de vezes tal que se obtenha o estado de equilíbrio do sistema, isto é, até que não haja melhora na função objetivo. A sequência de temperaturas e de configurações  $\{x_i\}$  dos átomos perturbados a cada temperatura representa o esquema geométrico do RS. Para entender mais sobre o RS veja Kirkpatrick et al. (1983) e suas referências, além de Rayward-Smith et al. (1996).

### 5.5.1 Descrição do Algoritmo RS

Nesta seção serão apresentados os passos da execução do algoritmo RS e será utilizada a seguinte notação:

- 
- i)*  $f$  é a função objetivo a ser minimizada;
  - ii)*  $x$  é a solução gerada na iteração corrente;
  - iii)*  $x^*$  é a melhor solução encontrada a uma dada temperatura;
  - iv)*  $T_0$  é a temperatura inicial e  $T$  é a temperatura corrente;
  - v)*  $\rho$  é um número, no intervalo  $[0, 1]$ , uniformemente distribuído.

Com a notação adotada acima, os passos do algoritmo SA são:

- *Passo 1* - Atribuir a  $x$  uma solução (configuração) inicial;
- *Passo 2* - Fazer  $x^* = x$ ;
- *Passo 3* - Definir uma temperatura inicial  $T_0$ . Fazer  $T = T_0$ ;
- *Passo 4* - Verificar se algum critério de parada é satisfeito;
- *Passo 5* - Escolher  $x'$  um ponto vizinho de  $x$ ;
- *Passo 6* - Calcular  $\Delta E = f(x') - f(x)$ ;
- *Passo 7* - Verificar se  $\Delta E < 0$ ;
- *Passo 8* - Caso o passo 7 seja satisfeito, faça:  $x^* = x'$ ;
- *Passo 9* - Caso o passo 7 não seja satisfeito, gerar aleatoriamente  $\rho \in [0, 1]$  com distribuição uniforme. Se  $\rho < P(\Delta E)$ , fazer  $x^* = x'$ ;
- *Passo 10* - Retornar ao passo 5;
- *Passo 11* - Atualizar a temperatura  $T$ ;
- *Passo 10* - Retornar ao passo 4.

## 5.6 Estudo de POMO via Métodos Heurísticos de Otimização

Conforme foi apresentado no capítulo 3, um POMO pode ser transformado em um problema mono-objetivo e as soluções ótimas do problema mono-objetivo são soluções ótimas de Pareto para o caso multiobjetivo. Porém, como são métodos ponto a ponto, necessitariam de muitas simulações do algoritmo para se obter a curva de Pareto. Além disso, neste contexto, não é assegurada uma boa distribuição das soluções sobre a curva de Pareto. Por outro lado, por trabalharem com populações e por permitirem a inserção de mecanismos que melhoram a diversidade, os métodos heurísticos constituem-se como estratégias mais adequadas para a resolução de POMO (LOBATO, 2008; DEB, 2001). Nesta seção serão apresentados, de forma bem sucinta, os principais mecanismos utilizados nos principais algoritmos encontrados na literatura para o estudo de POMO via métodos heurísticos. Ao final do capítulo será apresentada uma tabela com alguns dos principais algoritmos multiobjetivo, seus mecanismos para convergência e diversidade, bem como suas vantagens e desvantagens.

Conforme pode ser observado na descrição das heurísticas deste capítulo, em todos os casos tratou-se, de forma implícita, o problema de otimização como sendo mono-objetivo. No caso mono-objetivo há apenas uma meta a ser obtida, ou seja, a convergência do algoritmo. Para o caso multiobjetivo, além da convergência, há outra meta a ser alcançada, a diversidade (distribuição) das soluções sobre a curva de Pareto.

A maioria dos principais algoritmos heurísticos de otimização multiobjetivo são baseados em AG. Sendo assim, nesta seção, algumas terminologias dos algoritmos genéticos serão utilizadas na descrição dos mecanismos citados.



### 5.6.1 Mecanismo de Avaliação da Aptidão

Em todos os métodos heurísticos descritos nas seções anteriores deste capítulo a avaliação da aptidão das soluções potenciais eram realizadas em cada geração e o mecanismo de seleção escolhia as soluções mais aptas, em detrimento às soluções menos aptas, para a geração seguinte. Sendo assim, uma nova forma de avaliação da aptidão deve ser definida para o caso multiobjetivos.

Os principais mecanismos de aptidão utilizam o critério de dominância de Pareto, porém há algoritmos que para avaliarem a aptidão fazem o uso da média ponderada dos objetivos normalizados e até mesmo algoritmos que não possuem tarefa de aptidão. Abaixo são descritos os principais mecanismos:

- **Ranking de Pareto** - Neste caso, para cada solução potencial  $x_i$  é associado um valor de *ranking*  $r(x_i)$  igual ao número de soluções que a dominam  $d_{x_i}$  mais um, isto é,

$$r(x_i) = 1 + d_{x_i} \quad (5.13)$$

Assim as todas as soluções não dominadas possuem *ranking* 1. Sempre há um indivíduo da população com *ranking* 1 e o valor máximo para o *ranking* é o tamanho da população. A Figura 5.7 abaixo, ilustra um conjunto de soluções e seus respectivos *rankings*.

- **Aptidão Compartilhada** - Este mecanismo de aptidão compartilhada atua tanto no sentido de avaliação da aptidão de soluções quanto na obtenção da diversidade das mesmas e utiliza o conceito de nichos para soluções com o mesmo Ranking de Pareto. Sejam  $x_i$  e  $x_j$  duas soluções potenciais. Para definir os nichos, Goldberg (1989)

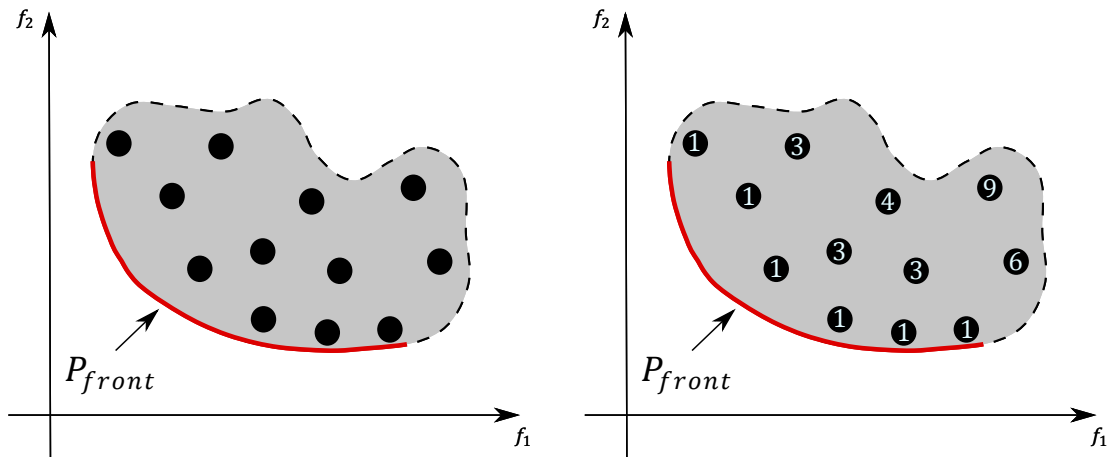


Figura 5.7: *Ranking* de soluções obtido pelo critério de dominância. Adaptada de (LOBATO, 2008).

utilizou a seguinte função:

$$Sh(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{share}}\right)^\alpha & \text{se } d_{ij} \leq \sigma_{share}; \\ 0 & \text{caso contrário.} \end{cases} \quad (5.14)$$

com

$$d_{ij} = \sqrt{\sum_{k=1}^m \left(\frac{f_k(x_i) - f_k(x_j)}{f_k^{max} - f_k^{min}}\right)^2} \quad (5.15)$$

onde o parâmetro  $\alpha$  define o comportamento da função  $Sh$  e  $\sigma_{share}$  é um valor definido pelo usuário e chamado de raio de nicho. Se a distância entre duas soluções for maior que o raio de nicho então elas estão em nichos separados. A Figura 5.8 ilustra um caso de soluções agrupadas em nichos.

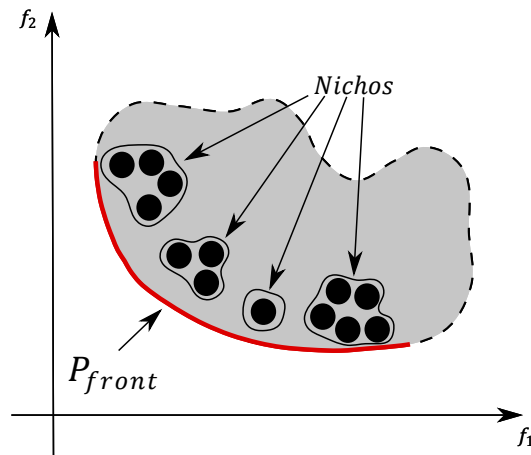


Figura 5.8: Soluções agrupadas em nichos. Adaptada de (DEB, 2001).

Considere ainda a seguinte função denominada de contador de nicho:

$$nc(x_i) = \sum_{j=1}^{\mu(x_i)} Sh(d_{ij}) \quad (5.16)$$

onde  $\mu(x_i)$  é o número de soluções de mesmo ranking de Pareto da solução  $x_i$ . Observe que quanto maior for o número de soluções de mesmo *ranking*  $r(x_i)$  maior é o valor de  $nc(x_i)$ .

Para definir a aptidão compartilhada de soluções no mesmo ranking, o valor do ranking de Pareto de uma solução  $x_i$  é dividido pelo valor contador de nicho  $nc(x_i)$ . Assim o valor de aptidão leva em consideração tanto o ranking de Pareto (baseado no critério de dominância) quanto o fato desta solução estar em um nicho menos ocupado.

- **Torneio de Pareto** - Durante o torneio, duas soluções  $x_i$  e  $x_j$  são escolhidas aleatoriamente na população  $P$ . Define-se um conjunto  $T$  tal que  $|T| < |P|$ . Pode acontecer as seguintes situações:

- 
- i)* Se a solução  $x_i$  domina o conjunto  $T$  e a solução  $x_j$  é dominada por alguma solução de  $T$ , então a solução  $x_i$  é a vencedora do torneio. Reciprocamente, se a solução  $x_i$  for dominada por alguma solução de  $T$  e a solução  $x_j$  dominar o conjunto  $T$ , então  $x_j$  é a solução vencedora;
- ii)* Se ambas soluções dominarem o conjunto  $T$ , ou se alguma solução de  $T$  dominar  $x_i$  e  $x_j$ , calcula-se o contador de nicho para determinar qual a solução vencedora do torneio.
- **Roleta** - De forma similar ao caso mono-objetivo, as soluções potenciais são ordenadas de acordo com o ranking de Pareto. Duas soluções  $x_i$  e  $x_j$  são escolhidas de forma aleatória na população. Pode-se ter as seguintes situações:

*i)* A solução  $x_i$  está melhor rankeada do que a solução  $x_j$ , então a solução  $x_i$  é a vencedora. Reciprocamente, a solução  $x_j$  possui melhor ranking do que a solução  $x_i$ , então  $x_j$  é a solução vencedora;

*ii)* Caso as soluções  $x_i$  e  $x_j$  possuam o mesmo ranking então calcula-se o contador de nicho para escolher qual a solução vencedora.
  - **Torneio de Multidão** - Há um operador, denominado distância de multidão a ser descrito na próxima seção, que é utilizado no método de seleção de soluções denominado Torneio de Multidão. Uma solução  $x_i$  é considerada vencedora do torneio contra uma solução  $x_j$ , quando (DEB, 2001):

*i)* A solução  $x_i$  está melhor rankeada do que a solução  $x_j$ , então a solução  $x_i$  é a vencedora;

*ii)* Caso as soluções  $x_i$  e  $x_j$  possuam o mesmo ranking de Pareto, porém a solução  $x_i$  possui distância de multidão maior do que a solução  $x_j$ .

## 5.6.2 Mecanismo de Corte de Soluções

Os algoritmos heurísticos de otimização trabalham com uma população de tamanho fixo. Após o processo de geração de novas soluções, faz-se necessário selecionar as melhores soluções para a geração seguinte. Assim algumas soluções devem ser eliminadas com base em algum critério. O principal critério utilizado para escolha de soluções que comporão as gerações seguintes baseia-se no critério de não dominância, isto é, uma solução com melhor nível de não dominância deve ser preferida em relação a uma com um nível de não dominância pior. Porém o que fazer quando as soluções em questão possuem o mesmo nível de dominância? A resposta para esta pergunta fundamenta-se em uma meta específica para POMO, a diversidade de soluções sobre a curva de Pareto.

O principal mecanismo de inserção/melhoria da diversidade das soluções sobre a curva de Pareto é fundamentado no operador denominado *distância de multidão* (*crowding distance*), introduzido no algoritmo NSGA II (DEB et al., 2000). Dada uma solução  $x_i$ , a distância de multidão  $d_i$ , da solução  $x_i$ , é uma estimativa do perímetro formado pelo cubóide cujos vértices são seus vizinhos mais próximos. Quanto maior for o perímetro, mais afastada a solução encontra-se de seus vizinhos e soluções extremas terão cubóide infinito. A Figura 5.9 abaixo ilustra a distância de multidão de uma solução  $x_i$ .

A utilização do operador de distância de multidão para o corte de soluções ocorre da seguinte forma. Suponha que em uma geração uma população  $P$  gerou uma população filha  $Q$ , ambas de mesmo tamanho e que o conjunto de soluções  $P \cup Q$  esteja ordenado pelo critério de não dominância. A fronteira  $F_1$  é formada pelas soluções de *ranking* 1, isto é, pelas soluções não dominadas. A fronteira  $F_2$  é formada pelas soluções que são dominadas apenas por alguma solução da fronteira  $F_1$ . A fronteira  $F_3$  é formada pelas soluções que são dominadas apenas por soluções das fronteiras  $F_1$  e  $F_2$  e assim sucessivamente. Cada conjunto  $F_i$  deve ser inserido, na sua totalidade, na composição da nova população  $P$

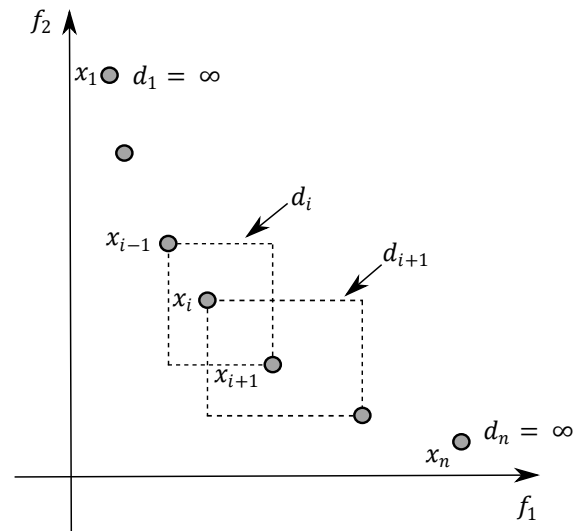


Figura 5.9: Representação do operador distância de multidão do NSGA II. Adaptada de (DEB, 2001).

(próxima geração) enquanto  $|P| + |F_i| \leq N$ , onde  $N$  é o tamanho da população. Ao inserir  $F_j$  tal que  $|P| + |F_j| > N$ , o algoritmo escolhe as soluções de  $F_j$  que estejam melhor espalhadas, isto é, aquelas com maior distância de multidão. A Figura 5.10 ilustra o mecanismo de corte de soluções realizado pelo algoritmo NSGA II.

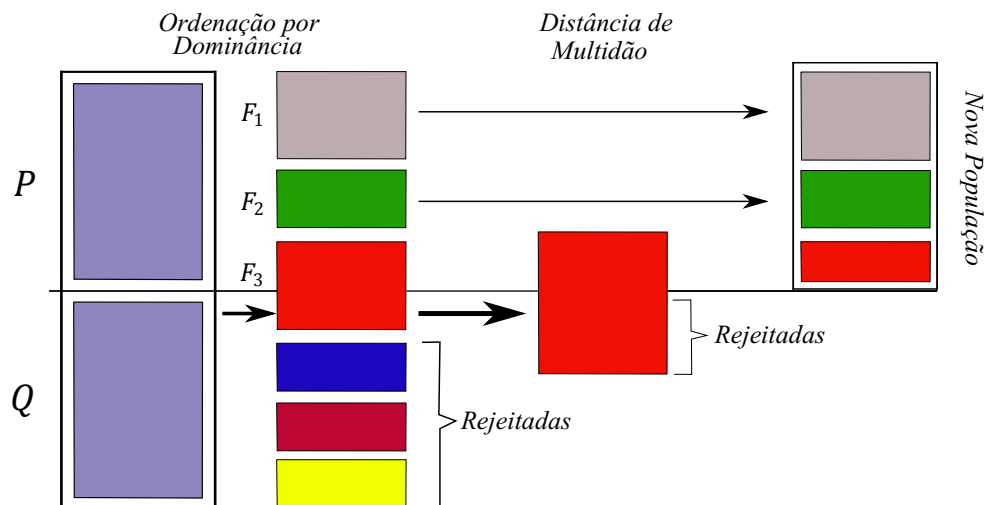


Figura 5.10: Representação do mecanismo de corte do NSGA II. Adaptada de (DEB, 2001).

Apesar de, na atualidade, este mecanismo de corte de soluções ser o mais utilizado em algoritmos multiobjetivo, um ponto negativo deve ser salientado. Suponha que a fronteira  $F_1$ , formada por soluções pais  $P$  e filhas  $Q$ , contenha mais elementos que a população inicial do algoritmo. O mecanismo de corte, para esta situação, com certeza eliminará soluções não dominadas, preservando aquelas com maior distância de multidão. Pode acontecer que na fronteira  $F_1$  existam soluções ótimas de Pareto muito próximas uma das outras e que haja também uma solução não ótima de Pareto, porém não dominada naquela geração. Se a distância de multidão desta solução em questão, for maior, então esta solução será copiada para a próxima geração enquanto que uma solução ótima de Pareto pode ser eliminada. Deb (2001) afirma porém que, apesar desta situação, o NSGA II cai em um ciclo de gerar soluções ótimas de Pareto e não ótimas de Pareto até convergir para um conjunto ótimo de Pareto. A Figura 5.11 a seguir ilustra a situação de uma possível eliminação de uma solução ótima de Pareto e manutenção de uma solução não dominada, porém não ótima de Pareto, para uma próxima geração do algoritmo.

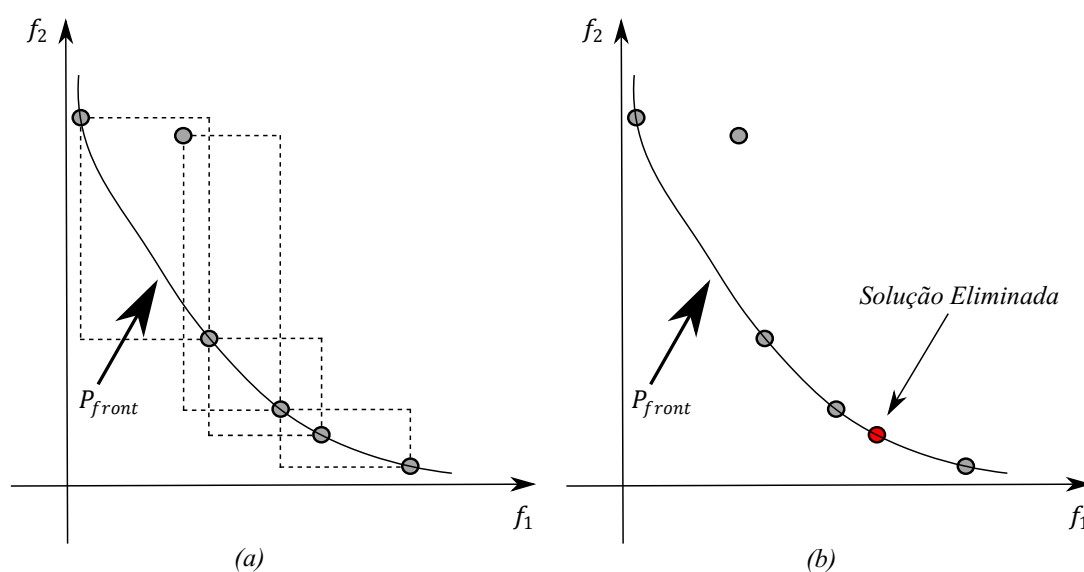


Figura 5.11: Situação de erro no mecanismo de corte por distância de multidão. Adaptada de (DEB, 2001).

A Tabela 5.2 lista alguns dos principais algoritmos multiobjetivo evolutivos da literatura, bem como suas tarefas de aptidão, mecanismo de diversidade, vantagens e desvantagens.



Tabela 5.2: Alguns dos Principais MOEA (*Multi-Objective Evolutionary Algorithms*) (LOBATO, 2008).

<i>Algoritmo</i>	<i>Tarefa de Aptidão</i>	<i>Mecanismo de Diversidade</i>	<i>Vantagens</i>	<i>Desvantagens</i>
<i>MOGA - Multi-objective Genetic Algorithm (FONSECA ; FLEMING, 1993)</i>	<i>Ranking de Pareto</i>	<i>Aptidão compartilhada por nicho</i>	<i>Extensão Simples de um AG para um objetivo</i>	<i>Baixa taxa de convergência e dependência do parâmetro de nicho</i>
<i>NPGA - Niche Pareto Genetic Algorithm (HORN et al., 1994)</i>	<i>Torneio de Pareto</i>	<i>Contador de nicho para soluções empatadas no torneio</i>	<i>Seleção bastante simples através do torneio</i>	<i>Dependência dos parâmetros: tamanho do torneio e tamanho de nicho</i>
<i>NSGA II - Non-Dominated Sorting Genetic Algorithm (DEB et al., 2000)</i>	<i>Ordenamento das soluções por não dominância</i>	<i>Distância de Multidão</i>	<i>Muito eficiente e bastante testada</i>	<i>A distância de multidão trabalha apenas no espaço de objetivos</i>
<i>SPEA II - Strength Pareto Evolutionary Algorithm (ZITZLER et al., 2001)</i>	<i>Baseado na Força dos dominadores</i>	<i>Densidade baseada na k-ésima vizinhança mais próxima</i>	<i>Pontos extremos não são perdidos</i>	<i>Computacionalmente caro</i>
<i>PMOGA - Pareto Multi-objective Optimization Genetic Algorithm (CASTRO, 2001)</i>	<i>Usa o conceito de dominância</i>	<i>Operadores de adição, exclusão e otimização individual</i>	<i>Rápida convergência e tratamento de problemas discretos</i>	<i>Não trabalha com codificação real</i>
<i>MODE - Multi-objective Differential Evolution (LOBATO, 2008)</i>	<i>Ordenamento das soluções por não dominância</i>	<i>Distância de Multidão</i>	<i>Trabalha com variáveis discretas e reais</i>	<i>Requer maior tempo de processamento quando comparado ao NSGA II</i>

# Capítulo 6

## Otimização Robusta

Conforme foi descrito anteriormente, existem problemas que dificultam a identificação de um ótimo para o problema de otimização, como por exemplo, a multimodalidade. Outras questões surgem naturalmente quando se deseja encontrar um ponto no espaço de projeto com uma alta precisão. Neste contexto, Leidemer (2009) ressalta alguns pontos, a saber:

- a) o ótimo verdadeiro talvez nunca possa ser possível de implementação prática, pois há incertezas associadas ao processo de construção. Além disso, a exigência de um alto grau de precisão na fabricação pode ser demasiadamente cara, portanto economicamente inviável;
- b) a formulação dos problemas de otimização são inerentemente estáticas enquanto a realidade é essencialmente dinâmica. Sendo assim, muitos problemas podem estar associados à flutuações de parâmetros tais como: temperatura, velocidade do vento, umidade ou mesmo pode haver desgaste de componentes.

Diante do exposto, os sistemas a serem otimizados podem ser bastante sensíveis a pequenas alterações das variáveis de projeto e assim pequenas variações nas variáveis

de projeto podem causar enormes variações dos objetivos (LEIDEMER, 2009), conforme argumentado anteriormente. Portanto, faz-se necessário encontrar uma metodologia que produza soluções que sejam pouco sensíveis a pequenas variações no projeto. Percebe-se através da Figura 6.1, que o objetivo  $f$  é muito sensível a pequenas variações em torno do ótimo global, e que o mesmo não ocorre numa vizinhança do ótimo local.

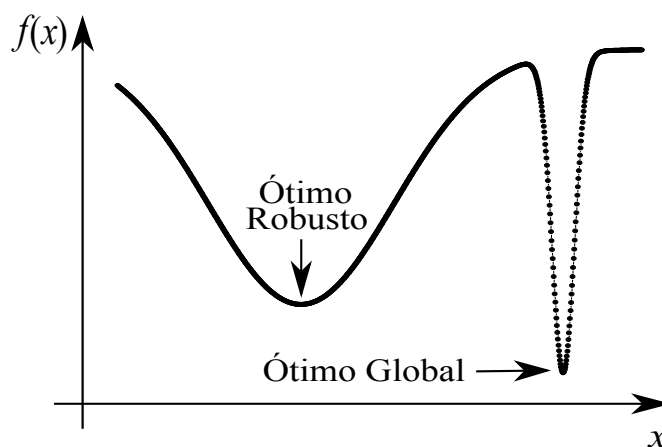


Figura 6.1: Ótimo global versus ótimo robusto. Adaptada de (DEB;GUPTA, 2006).

No contexto descrito acima, soluções pouco sensíveis a pequenas variações nas variáveis e/ou parâmetros são chamadas de robustas e o procedimento para encontrá-las é denominado Otimização Robusta. As origens da otimização robusta estão na engenharia e são diretamente relacionadas ao nome G. Taguchi, considerado o precursor desta área (TAGUCHI, 1984). O advento de computadores com alta velocidade e grande capacidade de realização de cálculos permitiu que houvesse um crescente interesse na otimização de projetos robustos nos últimos anos (DU;WANG;CHEN, 2000; BRANKE, 2002, PARK et al., 2006; BEYER;SENDHOFF, 2007).

Conforme Bennet (1990) alguns autores propõem a avaliação da robustez das soluções ótimas apenas ao final do processo de otimização, isto é, avaliam a sensibilidade dos objetivos e restrições ao perturbar uma solução ótima. Porém o uso desta estratégia requer

o conhecimento das funções envolvidas no processo. Outro ponto negativo desta estratégia é que apenas encontram-se zonas de robustez e não soluções robustas propriamente ditas, uma vez que soluções ótimas globais não são necessariamente robustas (LIMA, 2007).

Neste capítulo serão descritas as principais fontes de incertezas em projetos de otimização bem como sobre os principais mecanismos de incorporação das incertezas ao projeto, sendo que o objetivo principal é a definição do Problema de Otimização Multi-objetivo Robusto que será utilizada nas aplicações em matemática e de engenharia nos próximos capítulos desta Tese.

## 6.1 Trabalhos Relacionados

Nesta seção são apresentadas algumas aplicações envolvendo otimização robusta. Não são definidos conceitos, apenas algumas das principais metodologias utilizadas no contexto de otimização robusta.

Considerado o precursor da Otimização Robusta em engenharia, G. Taguchi estudou incertezas no âmbito da engenharia de qualidade (TAGUCHI, 1984). A metodologia proposta por Taguchi considerava a inserção, na função objetivo  $f$ , de fatores denominados por ele de fatores de ruído  $\xi$ . Mantendo  $x$  constante e considerando  $y_i = f(x + \xi_i)$  o valor do objetivo de uma amostra simples, Taguchi definiu duas funções, denominadas de Desvio Médio Quadrático, através das expressões:

$$\text{MSD}_1 = \frac{1}{k} \sum_{i=1}^k (y_i - \hat{y})^2 \quad \text{e} \quad \text{MSD}_2 := \frac{1}{k} \sum_{i=1}^k (y_i)^{-2} \quad (6.1)$$

onde  $\hat{y}$  é um valor alvo desejado e  $k$  é o tamanho da amostra. A função  $\text{MSD}_2$  era utilizada quando desejava-se que o objetivo  $f$  fosse o maior possível. Usando as funções

---

MSD, Taguchi definiu a função

$$\text{SNR} := -10 \log_{10}(\text{MSD}) \quad (6.2)$$

que deveria ser maximizada com respeito a  $x$ .

Conforme Beyer e Sendhoff (2007), Taguchi não utilizou nenhum algoritmo de otimização para maximizar a função definida na Eq. 6.2. Ao invés de um procedimento automatizado, Taguchi definiu uma matriz, denominada por ele de matriz de controle, a partir de valores amostrados para as variáveis de projeto  $x$  e de uma amostra para os fatores de ruído  $\xi$ . Para cada  $x$ , calculava-se o conjunto de valores  $\{y_1, y_2, \dots, y_k\}$  a partir do vetor de ruídos  $(\xi_1, \xi_2, \dots, \xi_k)$ , e então calculava-se os valores de  $\text{MSD}(x)$  e  $\text{SNR}(x)$ . Através de uma análise estatística dos dados era então identificado qual valor da variável de projeto  $x$  era capaz de produzir a melhor performance do objetivo.

Do ponto de vista prático, além do custo devido à quantidade de experimentos realizados, a metodologia proposta por Taguchi é claramente ineficiente quando o problema considerado possui um número elevado de variáveis de projeto (BEYER;SENDHOFF, 2007).

Os trabalhos de Park et al. (2006) e Beyer e Sendhoff (2007) revisam o estado da arte de projetos de otimização robusta mono-objetivo. Nestes trabalhos os autores definem os principais conceitos, as principais formas de inserção da robustez ao projeto, bem como as principais formas de tratamento de projetos robustos.

Parmee (1996) propôs um método de busca por soluções robustas que consiste na preferência de soluções, em regiões de alta performance do objetivo, que sejam do tipo “Platô” em detrimento às regiões do tipo “Pico”. Através desta preferência, o método proposto se restringe a estas regiões mais promissoras em termos de robustez. Porém,

---

como mostram Wiesmann et al. (1998), a solução ótima robusta não necessariamente está em uma região de alta performance do objetivo.

A metodologia de incorporação de incertezas (medidas de robustez) ao problema de otimização que é mais comum na literatura consiste no cálculo de integrais sobre os intervalos de variação dos parâmetros de incerteza e levam em conta a distribuição estatística dos parâmetros de incerteza. Estas medidas são denominadas medidas de expectância e de variância (DEB;GUPTA, 2006; JIN;SENDHOFF, 2003; PAENK et al., 2006). Como afirmam Paenk et al. (2006), estas medidas de robustez não podem ser calculadas analiticamente e, alternativamente, as medidas podem ser estimadas através da amostragem de Monte Carlo (METROPOLIS; ULAM, 1949). Porém o Método de Monte Carlo onera muito o custo computacional quando o problema possui muitas variáveis de projeto, implicando assim na estimativa das integrais tornar-se demasiadamente demorada. Então se faz necessária a utilização de uma técnica de amostragem mais eficiente no quesito tempo e que forneça uma boa estimativa para a integral. Segundo Paenk et al. (2006) um dos principais esforços de pesquisas em otimização robusta está em reduzir o esforço computacional na estimativa das medidas de robustez. Uma técnica que tem sido utilizada como alternativa ao método de Monte Carlo é o método denominado Hipercubo Latino, uma variante do método de Monte Carlo (EGLAJS;AUDZE,1977; VIANA et al., 2010). Esta abordagem de incorporação das incertezas tem sido utilizada com sucesso por pesquisadores em diversas aplicações. Paenk et al. (2006) utilizam a metodologia mencionada para estudar as soluções robustas de certos problemas matemáticos. Lima (2007) usou a mesma metodologia associada ao algoritmo NSGA II (*Nondominated Sorting Genetic Algorithms* (SRINIVAS;DEB, 1994)) para a otimização multiobjetivo robusta de sistemas mecânicos na presença de amortecimento viscoelástico. Borges (2008) usou a metodologia acima para estudar sobre a otimização robusta de absorvedores dinâmicos de vibrações.

Jin e Sendhoff (2003) consideram que todo problema de otimização robusta deve ser multiobjetivo mesmo quando o problema for mono-objetivo. Neste caso, tanto o objetivo original quanto seu equivalente robusto devem ser otimizados simultaneamente, buscando assim soluções que visam equilibrar a robustez e a performance do objetivo, que são objetivos conflitantes.

Segundo Paenk et. al (2006), de longe a maioria das atividades de pesquisa em otimização robusta se trata de problemas mono-objetivo. Também corroboram com esta afirmação alguns pesquisadores com atividade reconhecida na área de otimização evolucionária multiobjetivo. Deb e Gupta (2006) afirmam que, considerando o que conhecem sobre o tema, não há um estudo sistemático introduzindo robustez na otimização multiobjetivo. Sendo assim, é de fundamental importância o desenvolvimento e testes de metodologias consistentes para tratamento de problemas de otimização multiobjetivo robusta. Portanto, a presente tese, cujo principal objetivo é desenvolver uma metodologia consistente para resolver problemas de otimização multiobjetivo robusta, se apresenta como um estudo fundamental para esta área que ainda carece de estudos sistemáticos.

## **6.2 Representação dos Parâmetros de Incerteza**

Em situações reais de projetos de sistemas em engenharia, muitas vezes não é suficiente que uma solução seja de alta qualidade do ponto de vista determinístico, pois o projeto pode exigir que a solução seja robusta. A exigência da robustez ao projeto se deve ao fato de que os objetivos podem ser muito sensíveis a pequenas variações das variáveis de projeto e/ou dos parâmetros dos objetivos, fazendo com que a solução ótima seja instável.

Pequenas perturbações das variáveis de projeto e/ou dos parâmetros que aparecem nos objetivos de algum problema de otimização podem ser causadas principalmente por:

- flutuações das condições no ambiente podem causar pequenas variações em algum parâmetro dos objetivos;
- dificuldades geradas devido a altos custos para produzir uma solução com exigência excessiva no projeto ou mesmo pela incapacidade de se construir o projeto nas devidas especificações;
- erros na modelagem dos objetivos através de dados experimentais.

Em projetos elétricos, circuitos são submetidos a uma ampla gama de condições ambientais como flutuações na temperatura (THOMSON, 1996). Em projetos de “pás de turbinas”, o equipamento deve trabalhar eficientemente sob várias condições ambientais, tais como variações de velocidade. Condições muito similares são impostas em projetos ótimos de aerofólios (RAY; TSAI, 2004; YAMAGUCHI; ARIMA, 2002). A Figura 6.2 ilustra sobre as fontes dos principais tipos de incertezas associados ao projeto de otimização de sistemas em engenharia.

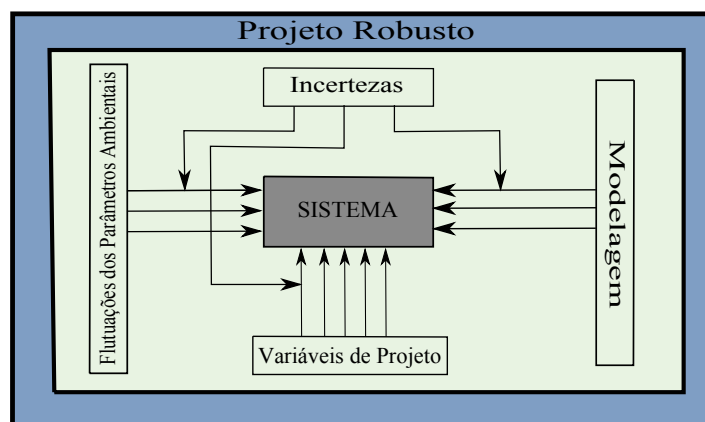


Figura 6.2: Principais fontes de incertezas em projetos. Adaptada de (BEYER; SENDHOFF, 2007).

Nesta tese serão estudados apenas problemas de otimização robusta com incertezas associadas às variáveis de projeto e/ou parâmetros. A quantificação destas incertezas



---

se dá através da inserção, nos objetivos e restrições, de novos parâmetros denominados *parâmetros de incerteza*. Considere um objetivo (o mesmo ocorre com uma restrição) genérico  $f : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$  representado por  $f(x, \alpha)$ , onde  $x$  é o vetor de variáveis de projeto e  $\alpha$  é o vetor de parâmetros do objetivo  $f$ . A inserção dos parâmetros de incerteza num objetivo  $f$  ocorre da seguinte forma (BEYER; SENDHOFF, 2007):

- *flutuações ambientais* - São associadas ao ambiente no qual o projeto está inserido. Pequenas perturbações em algum parâmetro do projeto podem causar grandes alterações nos valores dos objetivos ou até mesmo fazer com que uma solução ótima viole alguma restrição. Estas incertezas “entram” no sistema através de um vetor  $\Delta\alpha$  de variáveis associadas aos parâmetros dos objetivos, da seguinte forma:

$$f = f(x, \alpha + \Delta\alpha) \tag{6.3}$$

- *produção de tolerâncias* - Em um projeto de sistemas em engenharia, este pode ser executado apenas com um certo grau de acurácia. Alta precisão pode ser demasiadamente cara e, portanto, um projeto menos sensível à produção de tolerâncias reduz custos. Este tipo de incertezas entram no sistema através de um vetor de perturbação  $\delta$  das variáveis de projeto, da forma:

$$f = f(x + \delta, \alpha) \tag{6.4}$$

Observe que os dois tipos de incertezas apresentados acima podem aparecer simultaneamente em um projeto de otimização. Neste caso sua representação ocorre da seguinte

forma:

$$f = f(x + \delta, \alpha + \Delta\alpha) \quad (6.5)$$

Segundo Beyer e Sendhohh (2007) existem diferentes possibilidades para quantificar as incertezas representadas acima. Em geral estas incertezas podem ser modeladas deterministicamente, probabilisticamente ou possibilisticamente:

- a modelagem determinística define domínios (intervalos) em que as incertezas  $\Delta\alpha$ ,  $\delta$ , etc, podem variar;
- a probabilística define medidas de probabilidade descrevendo a possibilidade de certo evento ocorrer; e
- o tipo possibilístico define medidas *Fuzzy* descrevendo a possibilidade de um evento ser plausível ou acreditável.

Nesta tese todos os esforços serão concentrados no estudo de problemas de otimização robusta multiobjetivo onde as incertezas se apresentarão na forma paramétrica, diretamente sobre as variáveis de projeto. Beyer e Sendhoff (2007) apresentam, de forma sucinta, uma descrição dos outros dois tipos de modelagem dos parâmetros de incerteza, enquanto Agarwal (2004) realiza, em sua tese de doutorado, um estudo detalhado sobre incertezas do tipo probabilístico e Klir e Filger (1998) descrevem detalhadamente sobre incertezas modeladas pelo tipo possibilístico.

A modelagem determinística é a mais utilizada em projetos de otimização em engenharia. Muitos autores inserem como hipótese adicional que os parâmetros de incerteza possuem distribuição estatística conhecida (BRANKE, 2002; JIN; SENDHOFF, 2003; PAENK et al., 2006). Porém esta hipótese deve ser feita com muito critério, pois a menos que

se tenha uma idéia clara sobre a distribuição dos parâmetros de incerteza, esta hipótese pode ser incorreta em várias situações enfrentadas por projetos de engenharia. Em um problema com mais de uma variável e/ou parâmetro, há ainda a necessidade de realizar hipóteses sobre a correlação entre os parâmetros de incerteza. Outro inconveniente desta hipótese é que se deve atribuir valores também para os parâmetros das distribuições estatísticas dos parâmetros de incerteza.

Conforme a metodologia apresentada por Deb e Gupta (2006), nesta tese será suposto que os parâmetros de incerteza variam uniformemente em um intervalo fechado e limitado.

Na seção 6.3 serão descritas as principais medidas de incorporação dos parâmetros de robustez ao projeto encontradas na literatura e na seção 6.4 apresenta-se a definição do problema de otimização robusto multiobjetivo que será utilizado nos estudos de casos dos capítulos posteriores.

## **6.3 Medidas de Robustez**

Nesta seção são apresentadas as principais medidas de incorporação dos parâmetros de incerteza ao projeto de otimização. Estas medidas são denominadas na literatura por *Medidas de Robustez*. As incertezas consideradas nesta tese são do tipo Flutuações de Parâmetros Ambientais e Produção de Tolerâncias. A definição de medidas de robustez para estes dois tipos de incertezas é inteiramente análoga para ambas as medidas.

### 6.3.1 Regularização Robusta

Dado um objetivo  $f : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$  a ser minimizado, a medida de robustez denominada Regularização Robusta é definida como (LEWIS, 2002):

$$\tilde{f}(x, \xi) = \sup_{\delta \in B(x, \xi)} f(\delta) \quad (6.6)$$

onde  $B(x, \xi)$  é o círculo de centro em  $x$  e raio  $\xi$ , isto é, uma pequena vizinhança do ponto  $x$  cujo tamanho depende apenas do parâmetro de regularização  $\xi$ . O processo de otimização continua com a minimização da função  $\tilde{f}(x, \xi)$ . Tem-se que:

$$\lim_{\xi \rightarrow 0} \tilde{f}(x, \xi) = f(x) \quad (6.7)$$

A Regularização Robusta definida acima, para incertezas do tipo Produção de Tolerâncias, pode ser, de forma inteiramente análoga, definida para incertezas do tipo Flutuações Ambientais. Esta forma de incorporação de incertezas ao processo de otimização, definida pela Eq. 6.6, apresenta o cenário de pior caso em considerar o valor máximo de  $f$  sobre uma vizinhança da variável de projeto  $x$ . El Ghaoui e Lebret (1997) utilizam a técnica da regularização robusta para encontrar soluções robustas para problemas de mínimos quadrados. Beyer e Sendhoff (2007) apresentam, através de um exemplo para uma função unidimensional, um estudo sobre o efeito da regularização robusta e do efeito do “tamanho da vizinhança”  $\xi$  sobre a solução robusta.

Percebe-se claramente que a forma de contabilização de incertezas no processo de otimização definido através da Eq. 6.6 é bastante ineficiente, pois para a maioria dos projetos de otimização de sistemas em engenharia, os objetivos e restrições são funções bastante complexas, não permitindo assim exibir a forma analítica para a função de robustez de-

finida pela Eq. 6.6. Além disso, a escolha de um raio  $\xi$  muito grande pode resultar em soluções robustas de baixa qualidade, tornando-as inúteis (BEYER;SENDHOFF, 2007).

### 6.3.2 Medidas de Expectância e Variância

Segundo Branke (2002), Paenk et al. (2006) e Deb e Gupta (2006) as principais medidas de robustez são baseadas no cálculo de uma integral sobre os parâmetros de incertezas associados às variáveis de projeto e aos parâmetros dos objetivos. São comumente denominadas de medidas de expectância, pois em muitos casos considera-se que os parâmetros de incerteza seguem alguma distribuição conhecida.

**Definição 6.1.** *Seja  $f : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$  um objetivo de algum problema de otimização. A medida de incorporação da robustez ao processo de otimização denominada **Medida de Expectância**, e denotada por  $f_{exp}$ , é definida pela seguinte expressão matemática:*

$$f_{exp}(x, \alpha, \delta, \Delta\alpha) = E[f] = \int f(x + \delta, \alpha + \Delta\alpha)p(\delta, \Delta\alpha)d\delta d\Delta\alpha \quad (6.8)$$

onde  $p(\delta, \Delta\alpha)$  é a função de densidade conjunta dos parâmetros de incerteza  $\delta, \Delta\alpha$ . No caso de independência dos parâmetros, a função de densidade conjunta é substituída pelo produto das funções densidade de probabilidade de cada parâmetro de incerteza.

É importante observar que o parâmetro  $\alpha$  pode ser considerado como uma variável adicional da função objetivo e, conseqüentemente,  $\Delta\alpha$  como uma perturbação (Incerteza) desta variável. Sendo assim, nas definições seguintes, serão consideradas apenas as variáveis de projeto  $x$  e os parâmetros de incerteza  $\delta$ , associados às variáveis de projeto.

Segundo Beyer e Sendhoff (2007) a medida de robustez definida pela Eq. 6.8 é uma medida que tem sido frequentemente utilizada na literatura. A integral definida na Eq. 6.8 pode ser considerada como uma média ponderada, onde os pesos são as funções densidade

de probabilidade dos parâmetros de incerteza. Observe ainda que a função de robustez  $f_{exp}$  possui características melhores que as características do objetivo original  $f$ . Por exemplo, a função  $f_{exp}$  é derivável em qualquer ponto onde  $f$  seja apenas contínua e é contínua em pontos onde  $f$  é descontínua (onde a descontinuidade não atrapalhe a existência da integral). O fato de  $f_{exp}$  ser derivável, nos pontos de continuidade de  $f$ , implica uma maior suavidade do gráfico de  $f_{exp}$ , impedindo a existência de picos muito estreitos, isto é, melhora a robustez das soluções ótimas.

Outras medidas de robustez podem ser adicionalmente utilizadas para definir aspectos de robustez em uma aplicação específica. Por exemplo, a procura de regiões do tipo platô motiva a introdução de medidas de dispersão. Uma medida de dispersão é definida abaixo.

**Definição 6.2.** *Seja  $f : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$  um objetivo de algum problema de otimização. A medida de incorporação da robustez ao processo de otimização denominada **Medida de Variância**, e denotada por  $f_{var}$ , é definida pela seguinte expressão matemática:*

$$f_{var}(x, \delta) = \int (f(x + \delta) - f(x))^2 p(\delta) d\delta \quad (6.9)$$

onde, de forma análoga à Eq. 6.8, a função  $p(\delta)$  é a função de densidade conjunta dos parâmetros de incerteza  $\delta$  associados às variáveis de projeto.

Considerando a seguinte equação

$$Var[f] = E[(f - E[f])^2] = E[f^2] - (E[f])^2 = E[f^2] - (f_{exp})^2 \quad (6.10)$$

onde

$$E[f^2] = \int (f(x + \delta))^2 p(\delta) d\delta \quad (6.11)$$

obtem-se uma alternativa à medida de dispersão definida pela Eq. 6.9.

A busca por soluções ótimas robustas aparece frequentemente como um problema de otimização multiobjetivo (JIN; SENDHOFF, 2003). A robustez e a qualidade (valor numérico) de um objetivo, em diversas situações, são conflitantes, isto é, o aumento de um promove a diminuição do outro. Sendo assim, alguns autores afirmam que deve-se buscar um equilíbrio entre a robustez e a qualidade dos objetivos. Jin e Sendhoff (2003) consideram que as medidas de expectância, medidas de variância e os objetivos originais, devem ser combinados para formar um novo problema de otimização, que neste caso, será multiobjetivo. Diferentes combinações são possíveis, a saber:

- minimização do objetivo original e da medida de expectância associada a ele (DAS, 2000; BORGES, 2008);
- minimização do objetivo original e da medida de variância associada a ele;
- minimização da expectância e da variância (CHEN et al., 1996).

### 6.3.3 Média Efetiva

Conforme citado na subseção 6.3.2, as funções mais utilizadas para incorporação de robustez ao processo de otimização são as Medidas de Expectância. Estas consideram que os parâmetros de incerteza, associados a produção de tolerâncias e/ou flutuações ambientais, possuem distribuição estatística conhecida. Geralmente a hipótese levantada é de que os parâmetros de incerteza possuem distribuição normal com média zero e desvio padrão pequeno arbitrado pelo usuário (JIN;SENDHOFF, 2003).

Uma alternativa à hipótese sobre o conhecimento da distribuição dos parâmetros de incerteza é supor que eles estão uniformemente distribuídos em um intervalo simétrico. Deb e Gupta (2006) definem uma medida de incorporação da robustez ao processo de

otimização denominada por eles de Média Efetiva (*Mean Effective*) e a utilizam para realizar estudos em problemas multiobjetivos com solução analítica conhecida. A definição matemática para a função Média Efetiva é dada a seguir.

**Definição 6.3.** *Seja  $f : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$  um objetivo de algum problema de otimização. A medida de incorporação da robustez ao processo de otimização denominada **Média Efetiva**, e denotada por  $f^{eff}$ , é definida pela seguinte expressão matemática:*

$$f^{eff}(x, \delta) = \frac{1}{|B_\delta(x)|} \int_{y \in B_\delta(x)} f(y) dy \quad (6.12)$$

onde  $B_\delta(x) = \{y = (y_1, \dots, y_n) ; y_i \in [x_i - \delta_i, x_i + \delta_i], i = 1, 2, \dots, n\}$  é uma  $\delta$ -vizinhança de um ponto  $x$  e  $|B_\delta(x)|$  é o hipervolume desta vizinhança. Observe que a integral, definida pela Eq. 6.12, é uma integral múltipla. A forma como se encontra apresentada é apenas para simplificar a notação. Por exemplo, considerando um objetivo com duas variáveis de projeto, a integral, definida pela Eq. 6.12, seria escrita da seguinte forma:

$$f^{eff}(x_1, x_2, \delta_1, \delta_2) = \frac{1}{(2\delta_1)(2\delta_2)} \int_{x_1 - \delta_1}^{x_1 + \delta_1} \int_{x_2 - \delta_2}^{x_2 + \delta_2} f(y, z) dy dz \quad (6.13)$$

As integrais que aparecem nas Eq. 6.8, 6.9 e 6.12 não podem ser analiticamente calculadas para a maioria dos casos reais em engenharia, sendo necessária uma avaliação apenas aproximada para estas medidas de robustez. Existem muitas técnicas numéricas para calcular uma estimativa para as integrais citadas, como por exemplo: fórmulas de quadratura de Gauss, Regra dos Trapézios e Regras de Simpson. Porém, todos métodos numéricos necessitam de uma amostragem de pontos sobre o domínio de integração.

Sendo assim, é inerente ao processo de otimização robusta o aumento do número de avaliações dos objetivos. Se uma amostra de tamanho  $H$  é gerada para estimar a



integral, então  $H$  avaliações adicionais dos objetivos serão necessários a cada geração na otimização robusta (DEB;GUPTA, 2006). Portanto, uma boa técnica de amostragem não é aquela que somente ajuda a estimar a integral, porém é aquela que mesmo uma pequena amostra pode ser utilizada para realizar uma boa estimativa. Um método de amostragem, variante do Método de Monte Carlo, que vem sendo amplamente usado como alternativa é o método denominado Hiper cubo Latino (VIANA, 2008; VIANA et al., 2010). Este último não depende do número de variáveis, mas somente do tamanho da amostra que deseja-se extrair. Jin e Sendhoff (2003) e Paenk et al. (2006) utilizam o Método de Monte Carlo para estimar a integral dada na Eq. 6.8 enquanto Deb e Gupta (2006) usam o método do Hiper cubo Latino para estimar a integral dada pela Eq. 6.12. A Figura 6.3 ilustra, para um caso com duas variáveis, uma amostra do Método de Monte Carlo e outra realizada pelo Método Hiper cubo Latino.

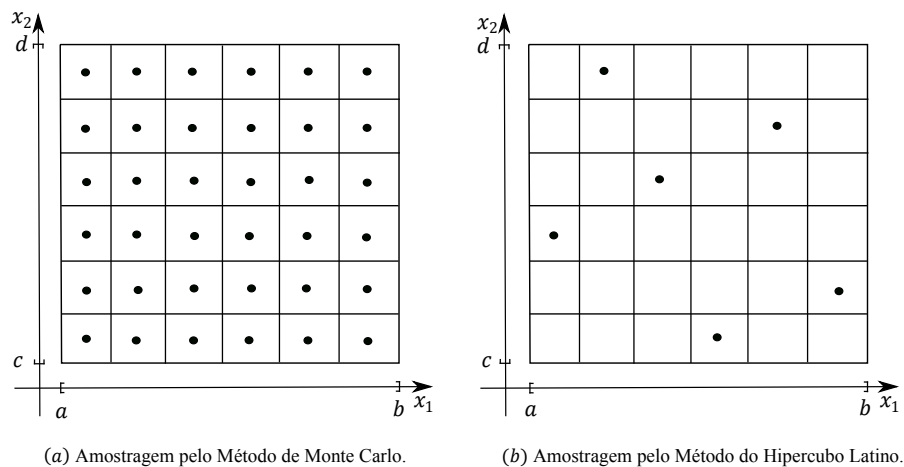


Figura 6.3: Amostragem pelos métodos de Monte Carlo e Hiper cubo Latino.

## 6.4 Otimização Robusta Multiobjetivo

No contexto de otimização multiobjetivo robusta, o estado da arte nos revela que somente alguns poucos trabalhos podem ser encontrados, dentre os quais Deb e Gupta (2006) e, mais recentemente, Moreira et al. (2013a, 2013b). Isto significa que são raros os trabalhos sistemáticos envolvendo a incorporação de robustez em problemas de otimização multiobjetivo aplicados a ciências e engenharia. Sendo assim, o objetivo principal desta seção é a conceituação de ótimo robusto para um problema de otimização multiobjetivo com restrições de desigualdade.

Conforme observa-se na Figura 6.1, apresentada anteriormente, uma pequena perturbação em torno da solução ótima produz uma grande perda de qualidade da função objetivo, isto é, a função objetivo é muito sensível a pequenas perturbações da solução ótima. Pode-se perceber que, em torno da solução robusta, o objetivo possui pequena sensibilidade e portanto pequenas perturbações numa vizinhança da solução robusta não deteriora a qualidade do objetivo. No que diz respeito ao atendimento das restrições de desigualdade, espera-se que as soluções robustas possam ser perturbadas sem se tornarem inviáveis. Observe que não faz sentido falar em robustez associada a problemas com restrições de igualdade, pois, certamente ao perturbar uma solução, este tipo de restrição seria automaticamente violada.

No contexto multiobjetivo define-se robustez de forma inteiramente análoga, porém agora a análise deve ser realizada sobre um conjunto de soluções igualmente ótimas, a Curva de Pareto. Note que se algum dos objetivos for sensível a pequenas perturbações numa vizinhança de uma solução, esta não será robusta, isto é, no contexto multiobjetivo a sensibilidade deve ser analisada considerando todos os objetivos simultaneamente. Conforme a Figura 6.4 ilustra, pequenas perturbações em uma vizinhança (no espaço de

variáveis) da solução  $B$  promove, no espaço de objetivos, grandes alterações nos valores dos objetivos. Analisando a solução  $A$ , percebe-se que os objetivos são pouco sensíveis a pequenas perturbações numa vizinhança desta solução. Neste caso, define-se a solução  $A$  como solução multiobjetivo robusta enquanto a solução  $B$  não é robusta.

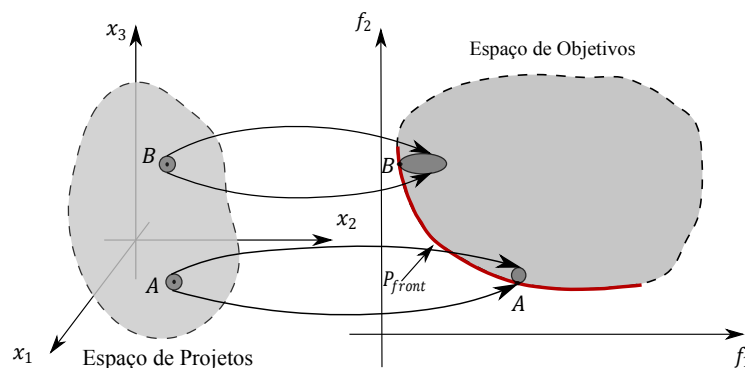


Figura 6.4: Sensibilidade de soluções em um espaço com dois objetivos. Adaptada de (DEB;GUPTA, 2006).

Em relação ao atendimento das restrições, soluções muito próximas à fronteira do espaço de variáveis de projetos devem ser preteridas em relação a soluções um pouco mais afastadas da fronteira. Isto é necessário, pois a robustez é avaliada sobre uma vizinhança de uma solução ótima de Pareto, que caso esteja muito próxima da fronteira, sua vizinhança pode conter pontos não viáveis, isto é, que não atendem às restrições.

O principal objetivo desta seção é a definição de solução multiobjetivo robusta bem como do Problema de Otimização Multiobjetivo Robusto (POMOR). Nesta tese utilizou-se o conceito de média efetiva, definido por Deb e Gupta (2006), para definir o POMOR.

Considerando o problema de otimização dado por:

$$\text{minimizar } \vec{F}(x) = (f_1(x), f_2(x), \dots, f_m(x)), \quad x \in \mathbb{R}^n \quad (6.14)$$

$$\text{sujeito a } \begin{cases} g_j(x) \leq 0, & j = 1, \dots, J \\ x_i^{inf} \leq x_i \leq x_i^{sup}, & i = 1, \dots, n. \end{cases} \quad (6.15)$$

conceitua-se solução multiobjetivo robusta. A solução de um Problema Multiobjetivo de Otimização Robusta é definida por:

**Definição 6.4.** *Uma solução  $x^*$  é denominada solução multiobjetivo robusta se for uma solução Ótima de Pareto para o seguinte problema de otimização multiobjetivo, definido em relação a uma  $\delta$ -vizinhança da variável de projeto  $x$ :*

$$\text{minimizar } \vec{F}(x) = (f_1^{eff}(x), f_2^{eff}(x), \dots, f_m^{eff}(x)), \quad x \in \mathbb{R}^n \quad (6.16)$$

$$\text{sujeito a } \begin{cases} g_j^{eff}(x) \leq 0, & j = 1, \dots, J \\ x_i^{inf} \leq x_i \leq x_i^{sup}, & i = 1, \dots, n. \end{cases} \quad (6.17)$$

onde o expoente *eff* nos objetivos e restrições significa a média efetiva destas funções definida através da Eq. 6.12. O problema definido nas Eq. 7.4 e 7.5, é denominado Problema de Otimização Multiobjetivo Robusto (POMOR).

**Definição 6.5.** *A Curva de Pareto Robusta é definida como sendo o conjunto de todos pontos ótimos de Pareto para o problema definido pela Eq. 7.4 e 7.5.*

Basicamente, no contexto multiobjetivo, pode-se observar três situações distintas para a Fronteira de Pareto Robusta (DEB;GUPTA, 2006):

- a Curva de Pareto é completamente robusta e, neste caso, a Curva de Pareto Robusta coincide com a Curva de Pareto Nominal, isto é, sem robustez;

- somente parte dos pontos sobre a Curva de Pareto são robustos e o restante dos pontos não são robustos. Neste caso, há uma parte da Curva de Pareto Robusta que coincide com a Nominal e outra parte que localiza-se próxima à Curva de Pareto Nominal;
- a Curva de Pareto Robusta é completamente distinta da Curva de Pareto Nominal.

Nos estudos de caso realizados no Capítulo 8 são apresentados casos onde as Curvas de Pareto Robustas ilustram a descrição acima.

# Capítulo 7

## Otimização Multiobjetivo via Colônia de Vagalumes

### 7.1 Aspectos Gerais do Algoritmo

Os métodos heurísticos evolutivos estão entre os mais utilizados para resolver problemas de otimização. Isto se deve, dentre outros aspectos, aos recursos computacionais hoje disponíveis, a capacidade de escapar de ótimos locais e a sua concepção conceitual simples. No entanto, nos últimos anos, algoritmos puramente estruturais, que simulam o comportamento social de animais ou fenômenos físicos têm sido amplamente utilizados no meio acadêmico (OLIVEIRA, 2006; LOBATO, 2008; SERAPIÃO, 2009; NETO; BECCENERI, 2009).

Algoritmos baseados no comportamento social de animais estão relacionados principalmente ao comportamento na busca por alimentos, características de aprendizagem, formas de aprendizagem e memorização e comportamento para acasalamento (SERAPIÃO, 2009). Em se tratando do comportamento coletivo de insetos, os vagalumes destacam-se

---

por sua característica bioluminescente de forma organizada e sincronizada, na busca por parceiros de acasalamento.

O pioneiro na implementação de um algoritmo de colônia de vagalumes foi o chinês Xin-She Yang (YANG, 2008). Na literatura podem-se encontrar algumas aplicações do ACV, mono e multiobjetivo, dentre as quais são citadas: sincronização de sensores em rede (Werner-Allen et al., 2005), projeto de redes sem fio (Leidenfrost e Elmenreich, 2008), otimização de funções matemáticas (Yang, 2008), resolução de um problema inverso de condução de calor Luz et al. (2009), desenvolvimento de uma estratégia auto-adaptativa usando modelos caóticos para a atualização dos parâmetros do ACV (Lobato e Steffen Jr, 2010), projeto de sistemas de engenharia (Lobato et al, 2011a), otimização de hidrociclones (Lobato et al., 2011b), estimação de parâmetros de controladores em processos químicos (Souza et al, 2012), controle ótimo (Lobato et al, 2012), dentre outras aplicações. Os trabalhos citados acima mostram que há um interesse da academia no estudo de algoritmos baseados nestes insetos.

No cenário da otimização robusta, a maioria das atividades de pesquisa em otimização robusta se trata de problemas mono-objetivo (PAENK et al. 2006; DEB;GUPTA, 2006). Neste contexto, objetivando contribuir com o estudo e pesquisa em otimização robusta multiobjetivo, é proposto o algoritmo MOFA (*Multiobjective Optimization Firefly Algorithm*), que é um algoritmo baseado no ACV para resolução de problemas de otimização multiobjetivo robustos. De maneira geral, na execução do MOFA, são empregados os seguintes operadores: estimativa dos objetivos definidos pela média efetiva (avaliação da integral), geração da população inicial, seleção e movimentação dos vagalumes, perturbação dos vagalumes, ordenamento elitista por *rank* e truncamento das soluções. Estes operadores serão descritos, com detalhes, nas próximas seções.

## 7.2 O Algoritmo MOFA

O usuário deve fornecer os seguintes parâmetros na execução do algoritmo MOFA:

- **Número de Objetivos** - quantidade de objetivos do problema, definido por  $N_{\text{obj}}$ ;
- **Número de Variáveis** - quantidade de variáveis de projeto do problema, definido por  $V$ , bem como seus respectivos domínios;
- **Número de Gerações** - usado como critério de parada, o número de gerações é representado por  $N_{\text{ger}}$ ;
- **Tamanho da Amostra** - usado na amostragem para obtenção de uma estimativa da média efetiva e é representada por  $N_{\text{amt}}$ ;
- **Tamanho da População** - define a quantidade de vagalumes utilizados no processo e é definido por  $N_{\text{vag}}$ ;
- **Coefficiente de Atratividade Máxima** - usado no cálculo da atratividade entre os vagalumes e é representado por  $\beta_0$ ;
- **Coefficiente de Absorção da Luz** - também usado no cálculo da atratividade entre os vagalumes e é representado por  $\gamma$ ;
- **Fator de Perturbação** - usado como critério para perturbar os vagalumes da colônia e representado por  $F_{\text{pert}}$ ;
- **Coefficiente de Robustez** - representa a amplitude da perturbação de cada variável de projeto  $x_i$  e é representado por  $\delta_i$ .

De forma bastante resumida, o algoritmo MOFA obedece à seguinte sequência de procedimentos:



- i. geração aleatória da população inicial;
- ii. realização da amostragem e cálculo de uma estimativa para os objetivos robustos;
- iii. classificação das soluções baseada no critério de não dominância por meio do *rank*;
- iv. para cada vagalume, da população  $P_1$ , calcular o vagalume que lhe é mais atrativo com base no *rank* e na distância entre os vagalumes;
- v. gerar outros  $N_{\text{vag}}$  através da movimentação dos vagalumes na direção do que lhe é mais atrativo e também através de um movimento aleatório. Este novos vagalumes são adicionados à população  $P_1$ , gerando uma população  $P_2$ . Se o critério, calculado com base nos vagalumes de dez gerações consecutivas, for maior que  $F_{\text{pert}}$  então é realizada a perturbação dos vagalumes em  $P_2$ ;
- vi. realizar a eliminação de  $N_{\text{vag}}$  vagalumes da população  $P_2$ , gerando uma nova população  $P_1$  com  $N_{\text{vag}}$  vagalumes, de forma que nenhuma solução não dominada seja perdida. Classificar as soluções por meio do *rank*.

Estes procedimentos são repetidos até que o número máximo de gerações  $N_{\text{ger}}$  seja alcançado. A saída do algoritmo MOFA é formada pelo conjunto de soluções não dominadas (variáveis do espaço de projetos) e seus respectivos valores dos objetivos. Nas próximas subseções são discutidos com mais detalhes os procedimentos do algoritmo MOFA.

### 7.2.1 Geração da População Inicial

Geralmente métodos heurísticos de otimização, baseados em processos evolutivos e em comportamentos coletivos de espécies, trabalham com uma população inicial e não somente com um ponto inicial. Sendo assim, o processo de inicialização destes algoritmos

consiste na geração de uma população inicial, obtida geralmente de forma aleatória. O processo de geração da população inicial depende do domínio de definição das variáveis de projeto.

Segundo Abbas et al. (2001) é possível encontrar na literatura outros métodos de geração da população inicial, que são baseados em distribuições estatísticas onde os parâmetros da distribuição são atribuídos pelo usuário.

No algoritmo MOFA a geração da população inicial é realizada de forma randômica. A geração de um indivíduo  $x = (x_1, x_2, \dots, x_n)$  pode ser realizada da seguinte forma:

$$x_i = x_i^{inf} + rand(x_i^{sup} - x_i^{inf}), \quad i = 1, \dots, n \quad (7.1)$$

onde *rand* é um número aleatório entre 0 e 1 e  $x_i^{inf}$  e  $x_i^{sup}$  são as restrições laterais da variável de projeto  $x_i$ .

## 7.2.2 Operador de Ordenamento

Na otimização mono-objetivo a avaliação da qualidade de um indivíduo da população é geralmente realizada pelo seu valor na função objetivo. Na otimização multiobjetivo o processo de avaliação de um indivíduo não pode ser realizado da mesma forma. No caso multiobjetivo as soluções não dominadas devem ser preferidas em relação às dominadas. Fonseca e Fleming (1993) propuseram uma forma de avaliação dos indivíduos baseada em um ordenamento por *ranking*, sendo que o *ranking* de um indivíduo é o número de soluções que a dominam mais um. Sendo assim, as soluções não dominadas possuem *ranking* 1.

O algoritmo MOFA realiza o mesmo ordenamento das soluções do algoritmo NSGA II (DEB et al., 2000). O ordenamento realizado pelo algoritmo NSGA II também é baseado no critério de não dominância onde as soluções são ordenadas da seguinte forma:

- as soluções não dominadas são classificadas como *rank 1* e retiradas da população;
- a população resultante é novamente classificada e os indivíduos não dominados são classificados como *rank 2* e depois são retirados da população;
- novamente a população resultante é classificada e os indivíduos não dominados são classificados como *rank 3* e retirados da população;
- o processo é repetido até que toda a população seja avaliada.

O processo de ordenamento por *rank*, descrito acima, é ilustrado na Fig. 7.1 a seguir.

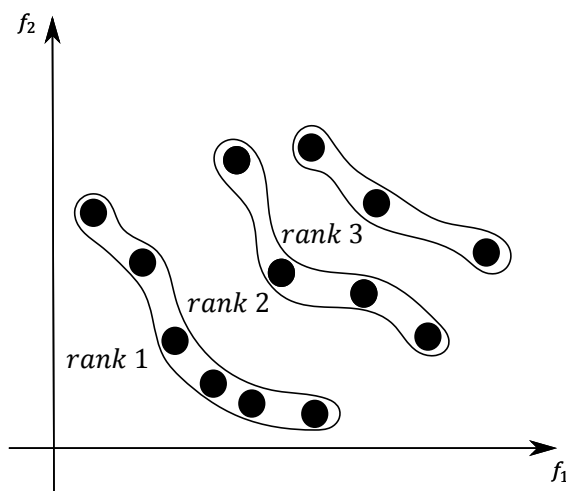


Figura 7.1: Ordenamento por *rank* realizado pelo algoritmo NSGA II. Adaptado de (DEB et al., 2000).

Segundo Silva (2004) o ordenamento das soluções realizado pelo NSGA II favorece a evolução das curvas e não de pontos individuais, facilitando a diversidade das soluções no espaço de objetivos. Este procedimento de ordenamento de soluções tem sido usado por vários autores na construção de algoritmos (CASTRO, 2001; SILVA, 2004; LOBATO, 2008; LOBATO et al., 2011a).

### 7.2.3 Operador de Truncamento

Cada indivíduo da colônia de vagalumes movimenta-se de acordo com a Eq. 5.9. Tanto os vagalumes de uma geração, quanto os obtidos pelos movimentos dos vagalumes da colônia, constituem uma nova população com o dobro do tamanho da população original. Sendo assim, para não onerar muito o custo computacional, deve ser incorporado um mecanismo de corte de soluções a cada geração. Inicialmente tal mecanismo objetiva o corte das soluções dominadas, mantendo as soluções não dominadas na população. A medida que aumenta o número de gerações, havendo a convergência do algoritmo, é natural de se esperar que o número de soluções não dominadas seja maior do que o tamanho da população inicial, que deve ser mantido fixo. Neste caso, o mecanismo deve eliminar também soluções não dominadas.

Segundo Deb (2001) não é interessante ter uma grande quantidade de indivíduos durante o processo e otimização, porém que os indivíduos estejam bem distribuídos ao longo da curva de Pareto. Sendo assim, visando atender a meta relativa à diversidade, soluções não dominadas melhor espaçadas devem ser preferidas em relação às soluções que mesmo não dominadas, estejam muito próximas de outras. O mecanismo de corte de soluções do MOFA é o mesmo do algoritmo NSGA II (DEB et al., 2000).

Este mecanismo é denominado na literatura como Distância de Multidão e foi apresentado na seção 5.6.2 do capítulo 5.

### 7.2.4 Operador de Seleção

Diferentemente dos algoritmos evolutivos, onde soluções são selecionadas para cruzamento, o ACV seleciona a melhor solução para que o indivíduo (vagalume) possa se dirigir em sua direção. A melhor solução depende do ponto de vista de cada indivíduo

e é função tanto da aptidão do vagalume observado, quanto da distância até o vagalume observador.

Dado um vagalume específico  $x_i$  é calculado o coeficiente de atratividade  $\beta$  para medir o quanto cada vagalume da colônia o atrai. Este coeficiente é função da aptidão (brilho emitido), baseada no *rank* de cada vagalume observado, e também da distância até o vagalume observador, visto que a intensidade do brilho emitido decresce com o aumento da distância. O vagalume com o maior coeficiente de atratividade é selecionado para que o vagalume  $x_i$  se movimente em sua direção segundo a Eq. 7.2, reproduzida a seguir

$$x_i^{k+1} = x_i^k + \beta(x_j^k - x_i^k) + \alpha \left( rand - \frac{1}{2} \right) \quad (7.2)$$

em que, o segundo termo da equação insere o fator de atratividade  $\beta$ , o terceiro termo, regulado pelo parâmetro  $\alpha$ , permite a inserção de certa aleatoriedade no caminho percorrido pelo vagalume e *rand* é um número aleatório entre 0 e 1. Observe que um vagalume não necessariamente segue o vagalume mais brilhante (melhor aptidão) da colônia e sim aquele que é mais atrativo a ele. Isto faz com que haja uma maior diversidade na população de vagalumes, pois se todos seguissem ao vagalume mais brilhante a população se estagnaria em poucas gerações (YANG, 2008).

### 7.2.5 Operador de Perturbação

O Operador de Perturbação, como o próprio nome diz, realiza pequenos deslocamentos nos vagalumes e tem como objetivo explorar a vizinhança dos vagalumes da colônia. Possui um efeito semelhante ao operador de mutação nos algoritmos evolutivos.

A perturbação dos vagalumes é realizada quando, em uma geração, a soma das médias

dos valores dos objetivos dos vagalumes da colônia apresenta um valor maior que uma amplitude pré-fixada quando comparado com o valor médio do mesmo número nas últimas dez gerações. O cálculo do critério, em cada geração  $i$  a partir da décima geração, é realizado da seguinte forma: (1) cada vagalume contribui com a média aritmética dos seus objetivos; (2) é realizada a soma das médias resultando no número  $soma - mediaobjetivos - vagalumes - i$ ; (3) a partir da décima geração é calculado, em valor absoluto, a diferença entre  $soma - mediaobjetivos - vagalumes - i$  e a média dos dez últimos valores para  $soma - mediaobjetivos - vagalumes$ , isto é, para  $soma - mediaobjetivos - vagalumes - (i - 10 : i)$ ; (4) se o número obtido em (3) for maior que a amplitude  $F_{pert}$  então é realizada a perturbação dos vagalumes. A perturbação é aplicada a todos vagalumes da população conforme a seguinte expressão:

$$\text{vagalume-novo} = 0,05 * (-1 + 2 * rand) * \text{vagalume-atual} \quad (7.3)$$

onde  $rand$  é um número aleatório entre 0 e 1.

Quando os vagalumes se aproximam da curva de Pareto então o processo de perturbação não é realizado, pois, não haverá uma diferença significativa na média dos objetivos com o passar das gerações.

## 7.2.6 Inserção do Parâmetro de Robustez - Aproximação da Média Efetiva

A inserção da robustez ao POMO realiza-se através da transformação do problema original, definido pelas Eq. 6.14 e 6.15, no problema robusto, definido pelas Eq. 7.4 e 7.5 que

são apresentadas novamente a seguir.

$$\text{minimizar } \vec{F}(x) = (f_1^{eff}(x), f_2^{eff}(x), \dots, f_m^{eff}(x)), \quad x \in \mathbb{R}^n \quad (7.4)$$

$$\text{sujeito a } \begin{cases} g_j^{eff}(x) \leq 0, & j = 1, \dots, J \\ x_i^{inf} \leq x_i \leq x_i^{sup}, & i = 1, \dots, n. \end{cases} \quad (7.5)$$

Para problemas complexos de engenharia não é possível realizar o cálculo analítico das médias efetivas, utilizadas na definição do POMOR. Sendo assim, o algoritmo MOFA realiza um procedimento de aproximação da integral definida na Eq. 6.12. O usuário deve informar as funções objetivo, as funções que definem as restrições e também a perturbação  $\delta_i$  em cada variável de projeto  $x_i$ . Observe que um parâmetro  $\alpha$  também está sendo considerado como uma nova variável de projeto, porém com valor fixo, isto é, o MOFA também considera a robustez associada a perturbações em parâmetros dos objetivos e/ou restrições. Para exemplificar como é realizada a entrada dos objetivos e restrições, considere o seguinte POMO:

$$\text{minimizar } \vec{F}(x) = (x_1^2, -x_2^3 + \cos(2x_2)), \quad x \in \mathbb{R}^2. \quad (7.6)$$

$$\text{sujeito a } \begin{cases} e^{x_1} - x_2^2 - 3 \leq 0, \\ -3 \leq x_i \leq 3, & i = 1, 2. \end{cases} \quad (7.7)$$

As perturbações das variáveis de projeto dos objetivos e a da restrição do POMO,

acima definido, devem ser inseridos no algoritmo MOFA da seguinte forma:

$$\text{minimizar } \vec{F}(x) = ((x_1 + \delta_1)^2, -(x_2 + \delta_2)^3 + \cos(2(x_2 + \delta_2))), \quad x \in \mathbb{R}^2 \quad (7.8)$$

$$\text{sujeito a } \begin{cases} e^{x_1 + \delta_1} - (x_2 + \delta_2)^2 - 3 \leq 0, \\ -3 \leq x_i \leq 3, \quad i = 1, 2. \end{cases} \quad (7.9)$$

Para o cálculo da aproximação da média efetiva, o usuário deve fornecer o valor, para cada variável  $x_i$ , de seu parâmetro de perturbação  $\delta_i$ , e o tamanho da amostra  $N_{\text{amt}}$ , que será calculada pelo método do Hipercubo Latino. Considerando uma função objetivo genérica  $f_i$ , o MOFA realiza a aproximação da média efetiva, através da seguinte expressão matemática:

$$f_i^{\text{eff}} \approx f_i^{\text{aprox}} = \frac{1}{N_{\text{amt}}} \sum_{j=1}^{N_{\text{amt}}} f_i(x_j + \delta_j) \quad (7.10)$$

Qualquer outro processo numérico para estimar a integral, definida na Eq. 6.12, poderia ser utilizado para esta finalidade.

## 7.2.7 Tratamento das Restrições

Na seção 4.6 foram apresentados dois métodos para o tratamento de restrições: a Penalização Externa e a Penalização Interna. Para descrever o tratamento das restrições em um POMO robusto, considera-se que há apenas restrições de desigualdade, pois restrições de igualdade certamente seriam violadas sob pequenas perturbações nas variáveis de projetos e/ou dos parâmetros das restrições.

Um dos parâmetros de entrada do usuário na otimização robusta é o tamanho da amostra  $N_{\text{amt}}$  usada para estimar a média efetiva. Observe na Figura 7.2 a seguir, que o



ponto  $A$ , apesar de ser viável, uma pequena perturbação em sua vizinhança pode conter pontos, obtidos a partir da amostragem realizada, que violam algumas das restrições.

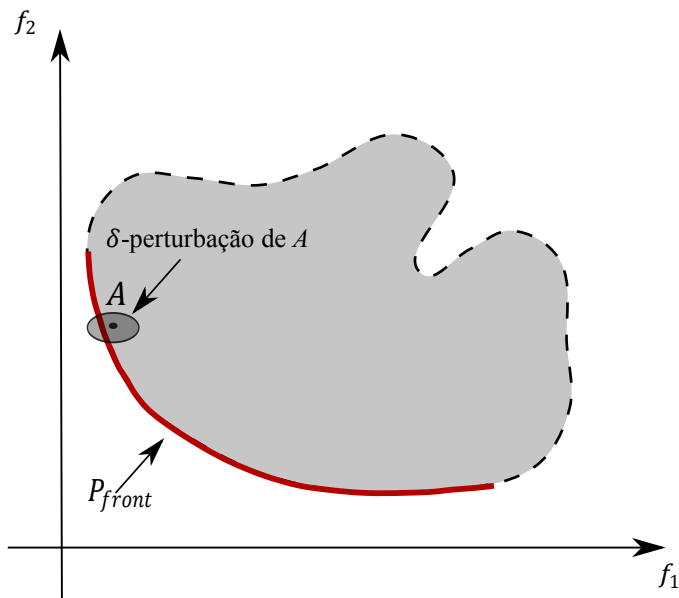


Figura 7.2: Análise da violação de restrições para o caso robusto devido a perturbação de uma solução viável.

O método para tratamento das restrições, usado pelo algoritmo MOFA, é definido através dos seguintes procedimentos:

- realiza-se a amostragem, através do método do Hipercubo Latino, em torno de um ponto genérico  $A$ ;
- lista-se, em um vetor, todas as violações de qualquer uma das restrições de desigualdade;
- soma-se a quantidade de violações das restrições, isto é, o tamanho do vetor de violações, gerando o número  $N_{viol}$ ;
- o número  $N_{viol}$  é multiplicado por um fator de penalidade  $r_p$  constante e o resultado é adicionado aos objetivos do POMO robusto;

Castro (2001) recomenda que os parâmetros de penalização sejam os valores limites de cada objetivo, pois, neste caso, é garantido que qualquer solução não dominada domine qualquer solução que apresente pelo menos uma violação de alguma restrição. Neste método de tratamento de restrições, o vetor de objetivos é escrito da seguinte forma:

$$f \equiv f + r_p N_{\text{viol}} \quad (7.11)$$

Neste contexto o método descrito acima pode ser considerado, na otimização robusta um método de penalização interna, pois mesmo no interior da região viável uma solução pode ser penalizada, isto é, quando uma amostra em sua vizinhança contiver pontos não viáveis. Observe que a função é penalizada se, e somente se, uma  $\delta$ -vizinhança de um ponto contiver pontos não viáveis, e neste caso, o número de violações  $N_{\text{viol}}$  é zero.

Para ilustrar uma aplicação do método de tratamento de restrições descrito nesta seção, considere um problema com três restrições de desigualdade. Suponha que um ponto viável contenha, em sua vizinhança de raio  $\delta$ , cinco pontos que violam a primeira restrição, dez que violam a segunda restrição e doze pontos que violam a terceira restrição. Neste caso o número de violações calculado é  $N_{\text{viol}} = 27$ .

### 7.2.8 Critério de Parada

Nesta tese o número máximo de gerações, definido antecipadamente pelo usuário, é utilizado como critério de parada do algoritmo MOFA. Este é sem dúvida o processo de parada mais utilizado nos algoritmos heurísticos de otimização multiobjetivo (DEB, 2001; AB-BASS et al., 2001, LOBATO, 2008; LOBATO et al., 2011a). Há outras formas de definir o critério de parada, a saber, as taxas de convergência definidas para cada objetivo. As taxas são definidas a partir das melhores soluções obtidas para cada objetivo. Quando

não há melhora das soluções para cada objetivo as taxas tendem a ser muito pequenas e, então, menores que um valor fixado a priori.

### 7.3 Estruturação do MOFA

A plataforma onde foi elaborado o algoritmo MOFA é o MATLAB<sup>®</sup>. O objetivo central desta seção é apresentar, de forma sucinta, as principais rotinas do algoritmo MOFA.

As principais rotinas do algoritmo MOFA são:

- **evaluate\_objective.m** - neste arquivo o usuário deve informar o tamanho da amostra ( $N_{\text{amt}}$ ) e os valores dos parâmetros de perturbação  $\delta_i$  de cada variável de projeto  $x_i$ . Este arquivo é responsável por realizar a estimativa da média efetiva dos objetivos;
- **evaluate\_sys.m** - neste arquivo o usuário deve entrar via teclado com os objetivos do problema robusto, conforme apresentado na subseção 7.2.6;
- **firefly.m** - este arquivo é responsável pela movimentação dos vagalumes e o usuário deve informar os valores dos parâmetros  $\beta_0$  e  $\gamma$ . Nele também é realizada a atualização dos valores dos objetivos dos novos vagalumes da população;
- **hypercube.m** - realiza o cálculo da amostra para posterior realização de uma estimativa para a média efetiva dos objetivos. O cálculo da amostra é realizado pelo método do Hiper cubo Latino (VIANA et al., 2010) e depende dos limites mínimo e máximo das variáveis de projeto e do tamanho da amostra, que é um dado informado pelo usuário;
- **initialize\_variables.m** - conforme o próprio nome indica, este arquivo é responsável pela geração da população inicial bem como sua avaliação nos objetivos do problema.

---

A geração da população inicial é realizada de forma aleatória, conforme a Eq. 7.1, e depende dos limites mínimo e máximo das variáveis de projeto;

- **metrics.m** - caso o problema possua solução analítica, a mesma é “carregada” na execução do arquivo `mofa.m` e então o cálculo das métricas de convergência (CM) e de diversidade (DM), cujas expressões são dadas pelas Eq. 8.1 e 8.2 respectivamente, é realizado através da execução do operador do arquivo `metrics.m`;
- **mofa.m** - é o arquivo principal do algoritmo MOFA. Neste arquivo o usuário deve informar, via teclado, os seguintes parâmetros: número de gerações ( $N_{ger}$ ), tamanho da população ( $N_{vag}$ ), Número de objetivos ( $N_{obj}$ ), os limites mínimo e máximo de cada variável de projeto e o fator de perturbação ( $F_{pert}$ ). No arquivo `moece.m` são acionadas as principais rotinas do algoritmo MOFA, é realizado o operador de perturbação dos vagalumes e, caso o problema possua solução analítica, então o operador que calcula os valores das métricas de convergência e de diversidade pode ser acionado;
- **non\_dominatation\_sort\_mod.m** - este arquivo é responsável pela distribuição das soluções em “pseudo-curvas”, baseadas nos “ranks” das soluções, conforme apresentado na subseção 7.2.2. Realiza também o cálculo da distância de multidão de cada vagalume da colônia, conforme apresentado na subseção 5.6.2 do capítulo 5;
- **replace\_firefly.m** - este arquivo é responsável pela substituição da população de vagalumes de uma geração pela população de vagalumes da geração seguinte. Após o algoritmo MOFA unir a população de vagalumes em uma geração com a população de vagalumes movidos, no arquivo `non_dominatation_sort_mod.m` é calculado o *rank* bem como a distância de multidão das soluções. Após estes cálculos o operador

do arquivo `replace_chromossome.m` realiza o truncamento das soluções conforme apresentado na subseção 7.2.3 deste capítulo e na subseção 5.6.2 do capítulo 5.

Como arquivos de saída, o algoritmo MOFA pode armazenar os seguintes dados:

- população inicial de vagalumes;
- valores do *rank* de cada vagalume da curva solução;
- distância de multidão para cada vagalume da curva solução;
- valores das variáveis de projeto de todos vagalumes pertencentes à curva solução;
- valores dos objetivos, avaliados sobre a curva solução;
- quando há solução analítica, o algoritmo MOFA, informa também os valores das métricas de convergência (CM) e de diversidade (DM).

A Figura 7.3 que se segue, ilustra a estruturação do algoritmo MOFA, através dos arquivos descritos anteriormente.

### *Multiobjective Optimization Firefly Algorithm*

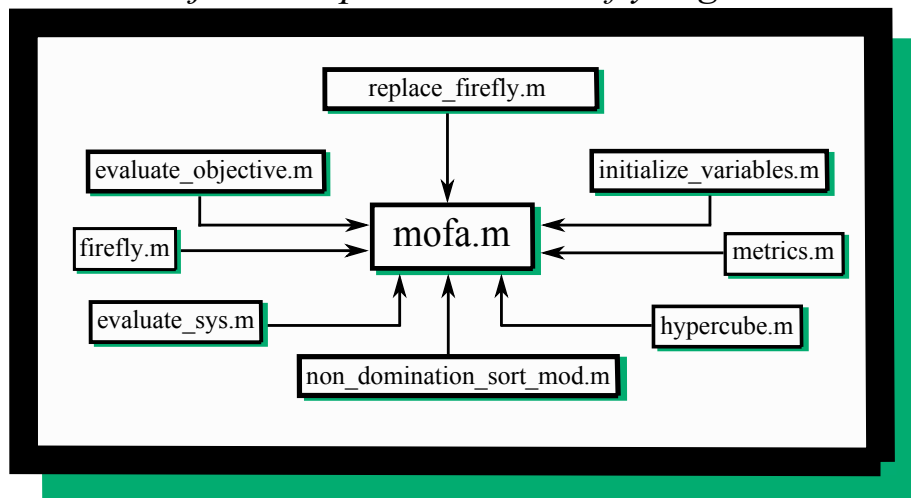


Figura 7.3: Arquivos componentes do algoritmo MOFA.

O algoritmo MOFA se constitui um algoritmo de otimização multiobjetivo robusta que pode ser considerado novo, pois há poucos trabalhos em termos de dissertações de mestrado e teses de doutorado que têm sido propostos com esta finalidade.

# Capítulo 8

## Aplicações Matemáticas

Neste capítulo são realizadas uma série de simulações do Algoritmo MOFA (*Multiobjective Optimization Firefly Algorithm*) em estudos de casos envolvendo problemas matemáticos. Objetivando a verificação da convergência do algoritmo MOFA e a diversidade das soluções, foram estudados problemas testes com soluções analíticas nominais e analíticas robustas conhecidas (problemas testes de 1 a 4). Foram realizadas discussões sobre a influência dos parâmetros de incerteza sobre o perfil da Curva de Pareto e da sensibilidade do algoritmo MOFA em relação a alguns parâmetros que devem ser informados pelo usuário.

Em geral, no Algoritmo de Colônia de Vagalumes (ACV), o usuário deve informar previamente os seguintes parâmetros: número de vagalumes  $N_{\text{vag}}$ , coeficiente de atratividade entre os vagalumes  $\beta_0$ , o coeficiente de absorção do brilho emitido  $\gamma$  e o número máximo de gerações  $N_{\text{ger}}$ . Para o caso robusto o usuário deve informar ainda o número de amostras necessárias para estimar a integral  $N_{\text{amt}}$  e o parâmetro de perturbação  $\delta$ . A Tabela 8.1 apresenta os valores dos parâmetros utilizados nas simulações realizadas pelo MOFA para os problemas teste deste capítulo.

Tabela 8.1: Parâmetros do algoritmo MOFA utilizados nas simulações.

<i>Parâmetros</i>	
$N_{\text{vag}}$	55
$N_{\text{ger}}$	250
$\beta_0$	1
$\gamma$	1
$N_{\text{amt}}$	100

No algoritmo MOFA o número de vagalumes é função do número de objetivos e do número de variáveis de projeto. Todos problemas estudados neste capítulo possuem dois objetivos e cinco variáveis de projeto e então o número de vagalumes é o mesmo nos problemas estudados. Foi realizada uma análise de sensibilidade dos parâmetros do algoritmo MOFA, apresentada em uma seção no final deste capítulo, e nela ficou evidenciado, para o problema estudado na análise, que os melhores resultados para a convergência foram obtidos quando  $\beta_0 = \gamma = 1$ . Para o cálculo de uma estimativa para a média efetiva, a análise de sensibilidade mostrou que os melhores resultados foram obtidos para um número de amostras a partir de cinquenta pontos. A escolha para o número de gerações se deu a partir da observação da convergência, em todos os casos estudados, para as curvas de Pareto dos problemas estudados.

Para avaliar a qualidade do algoritmo MOFA, as métricas de convergência (CM) e de diversidade (DM) foram utilizadas (DEB, 2001):

$$\text{CM} = \frac{\sum d_i}{N_{\text{vag}}} \quad (8.1)$$

$$\text{DM} = \frac{d_f + d_l + \sum |d_i - d_m|}{d_f + d_l + (N_{\text{vag}} - 1)d_m} \quad (8.2)$$

onde  $d_i$  é a distância euclideana entre o vagalume  $x_i$  e o ponto da Curva de Pareto mais



próximo de  $x_i$ ,  $d_f$  e  $d_l$  representam a distância euclideana entre as soluções extremas da curva de Pareto e do conjunto de soluções (vagalumes) não dominadas obtidas pelo algoritmo e  $d_m$  é a média aritmética das distâncias  $d_i$ . Para o cálculo das métricas CM e DM as curvas de Pareto analíticas (nominais e robustas) foram obtidas utilizando 1000 pontos.

Os objetivos principais para a realização dos estudos de casos contidos neste capítulo são:

- i*) estudar a convergência do algoritmo MOFA e a diversidade das soluções para os casos nominal e robusto;
- ii*) entender a influência do parâmetro de incerteza  $\delta$  na configuração das curvas de Pareto Robustas.

## 8.1 Problema Teste 1

Considere o problema definido por (DEB;GUPTA 2006):

$$\text{minimizar } \vec{F}_1(x) = (f_1(x), f_2(x)) = (x_1, h(x_1) + g(x)s(x_1)) \quad (8.3)$$

$$\begin{cases} h(x_1) = 1 - x_1^2 \\ g(x) = \sum_{i=2}^n 10 + x_i^2 - 10 \cos(4\pi x_i) \\ s(x_1) = \frac{\alpha}{0,2 + x_1} + \theta x_1^2 \\ 0 \leq x_1 \leq 1, \quad -1 \leq x_i \leq 1 \quad i = 2, \dots, n. \end{cases} \quad (8.4)$$

Para o estudo do problema teste 1 foram utilizados os valores  $\alpha = 1$  e  $\theta = 1$ . A curva de Pareto nominal para o problema  $F_1$ , definido pelas equações 8.3 e 8.4, corresponde a

$x_1 \in [0, 1]$  e  $x_i = 0$ ,  $i = 2, \dots, n$ . Observa-se que, sobre a curva de Pareto, a função  $g(x)$  é identicamente nula e, portanto, pode-se obter uma relação entre os objetivos  $f_1$  e  $f_2$ . Esta relação é expressa matematicamente por:

$$f_2 = 1 - f_1^2. \quad (8.5)$$

Nota-se que a geometria do problema (não convexo) no espaço de objetivos, entendida através de uma simples análise da Eq. 8.5, e o fato da função  $g$  ser multimodal causam dificuldade de convergência para qualquer algoritmo de otimização.

Para este estudo de caso a curva de Pareto robusta pode ser obtida analiticamente, mediante o cálculo das médias efetivas para os objetivos  $f_1$  e  $f_2$  em uma  $\delta$ -vizinhança do vetor de projeto  $x$ , sendo que esta vizinhança é obtida perturbando a coordenada  $x_i$  de  $x$  da forma  $[x_i - \delta_i, x_i + \delta_i]$ . Os objetivos obtidos pelas médias efetivas, avaliados sobre o conjunto de Pareto (robusto), são expressos por:

$$f_1^{eff}(x, \delta) = x_1 \quad (8.6)$$

$$f_2^{eff}(x, \delta) = 1 - x_1^2 - \frac{\delta_1^2}{3} + \left[ \frac{1}{2\delta_1} \ln \left( \frac{0.2 + x_1 + \delta_1}{0.2 + x_1 - \delta_1} \right) + x_1^2 + \frac{\delta_1^2}{3} \right] G(x) \quad (8.7)$$

onde

$$G(x) = \sum_{i=2}^n 10 + \frac{\delta_i^2}{3} - \frac{10}{4\pi\delta_i} \sin 4\pi\delta_i. \quad (8.8)$$

A curva de Pareto robusta, dada em função de  $\delta$ , pode ser obtida substituindo  $f_1^{eff}$

no lugar de  $x_1$  na Eq. 8.7 e fazendo  $x_1$  variar no intervalo  $[0, 1]$ . É interessante notar que, fazendo o parâmetro de perturbação  $\delta$  tender a zero, a curva de Pareto robusta tende à curva de Pareto nominal, pois,

$$\lim_{\delta \rightarrow 0} G(x) = \lim_{\delta \rightarrow 0} \left[ \sum_{i=2}^n 10 + \frac{\delta_i^2}{3} - \frac{10}{4\pi\delta_i} \sin 4\pi\delta_i \right] = 0, \quad (8.9)$$

e, portanto

$$\lim_{\delta \rightarrow 0} f_2^{eff}(x, \delta) = 1 - \left( f_1^{eff} \right)^2. \quad (8.10)$$

Os parâmetros do algoritmo MOFA, usados nas simulações, foram apresentados na Tab. 8.1. É importante ressaltar que para o caso nominal, isto é, sem robustez, são necessárias  $55 + 55 \times 250$  avaliações da função objetivo em cada execução. Já para o caso robusto, são necessárias  $55 + 55 \times 250 \times 100$  avaliações da função objetivo, pois, em cada geração são necessárias 100 avaliações adicionais do objetivo para estimar a integral. Neste caso, observa-se que há um grande acréscimo no número de avaliações para o caso robusto, como já era de se esperar.

Para fins de avaliação da convergência do algoritmo para os casos nominal e robusto, realizou-se dez simulações com o seguinte vetor de sementes iniciais para o gerador de números aleatórios *rand* do *software* Matlab<sup>®</sup> ( $[0 \ 1 \ \dots \ 9]$ ) e estipulou-se o número de variáveis  $n = 5$  (para todos Problemas Testes deste capítulo). Os desvios para cada uma das 5 variáveis de projeto foram definidos como  $[\delta_1 \ \delta_2 \ \delta_3 \ \delta_4 \ \delta_5] = [\delta \ 2\delta \ 2\delta \ 2\delta \ 2\delta]$  (DEB;GUPTA, 2006).

A Figura 8.1 apresenta a curva solução obtida pelo MOFA juntamente com a curva de Pareto do problema nominal. Observa-se graficamente a convergência do algoritmo MOFA e uma boa distribuição das soluções sobre a curva de Pareto.

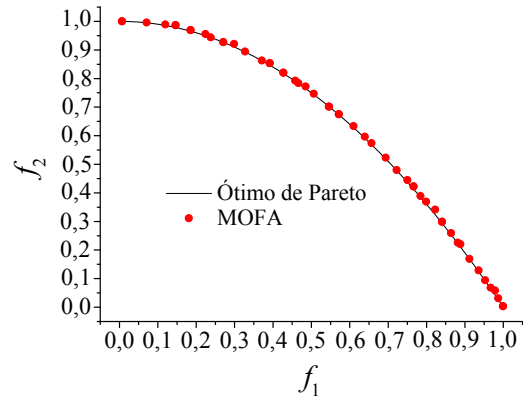


Figura 8.1: Curva de Pareto nominal e obtida pelo MOFA para o problema teste 1.

A Tabela 8.2 apresenta os resultados das métricas de convergência e de diversidade, obtidos nas dez simulações, para o problema teste 1 nominal. Observa-se pelos resultados da métrica CM que, em todas dez simulações, houve a convergência do algoritmo MOFA para a solução analítica do problema. Os resultados da métrica DM na Tab. 8.2 mostram que as soluções estão bem distribuídas, isto é, há uma boa diversidade das soluções.

Tabela 8.2: Resultados das métricas CM e DM na simulação do problema teste 1 nominal.

Métricas	Simulações									
	1	2	3	4	5	6	7	8	9	10
CM	0,003	0,003	0,003	0,002	0,004	0,002	0,002	0,003	0,002	0,003
DM	0,855	0,975	0,806	0,828	0,812	0,640	0,938	0,810	0,872	1,021

Para o estudo da robustez no problema teste 1 foram utilizados quatro valores para o parâmetro de incerteza  $\delta$ . Na Figura 8.2(a) estão contidas as curvas de Pareto robustas analíticas obtidas das equações 8.6 e 8.7 para os valores de  $\delta$ . No que diz respeito à sensibilidade dos objetivos, observa-se que para valores pequenos da variável  $x_1$  há uma maior sensibilidade do objetivo  $f_2^{eff}$ . A Figura 8.2(b) apresenta os resultados obtidos pelo algoritmo MOFA em função do parâmetro de incerteza  $\delta$ . Observa-se, graficamente, que há convergência do algoritmo MOFA, para cada um dos valores de  $\delta$ , para a solução

analítica do problema. Observa-se claramente nesta figura o efeito do acréscimo de inserção de incerteza no vetor de variáveis de projeto, ressaltando a importância deste efeito durante a otimização, através da mudança do perfil da curva de Pareto.

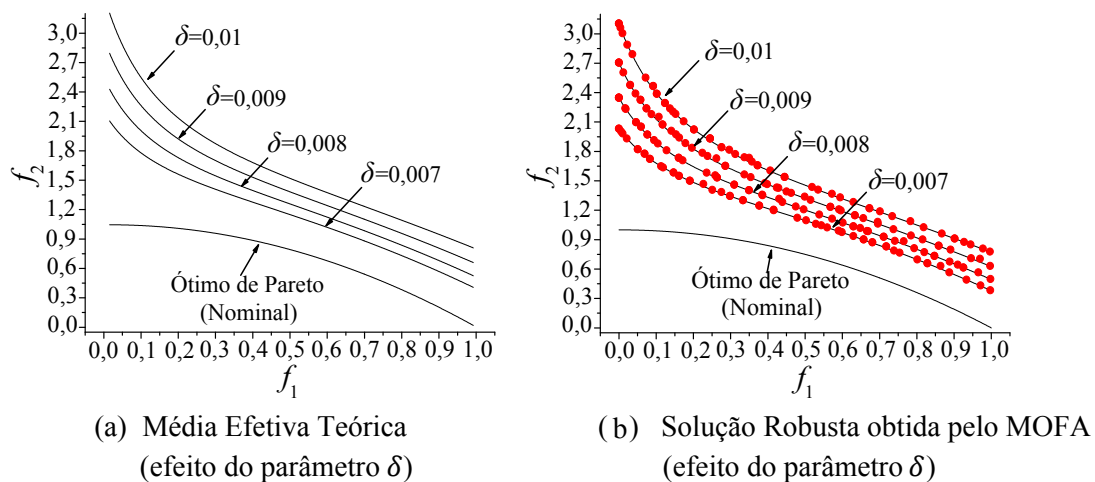


Figura 8.2: Curvas de Pareto robustas para o problema teste 1.

Para finalizar o estudo da convergência do algoritmo MOFA para o caso robusto, a Tab. 8.3 apresenta os resultados das métricas de convergência CM e de diversidade DM para os diferentes valores de  $\delta$  utilizados nas simulações. Pode-se então confirmar que o algoritmo MOFA convergiu satisfatoriamente para a solução analítica robusta para todos os valores do parâmetro de incerteza considerados.

Tabela 8.3: Resultados das métricas CM e DM na simulação do problema teste 1 robusto.

<i>Simulações</i>											
$\delta$	<i>Métricas</i>	1	2	3	4	5	6	7	8	9	10
0,01	CM	0,003	0,003	0,002	0,002	0,004	0,003	0,002	0,002	0,001	0,003
	DM	0,844	0,871	0,868	0,817	0,823	0,828	0,926	0,868	0,741	0,803
0,009	CM	0,003	0,002	0,005	0,002	0,004	0,003	0,003	0,003	0,004	0,003
	DM	0,791	0,769	0,924	0,950	0,964	0,825	0,921	0,985	0,894	0,956
0,008	CM	0,002	0,002	0,003	0,003	0,003	0,001	0,002	0,004	0,003	0,003
	DM	0,832	0,879	0,838	0,945	1,067	0,989	0,856	0,972	0,940	0,859
0,007	CM	0,002	0,003	0,003	0,005	0,003	0,002	0,003	0,003	0,003	0,003
	DM	0,834	0,741	1,056	0,844	0,900	0,654	0,890	0,780	0,896	1,053

## 8.2 Problema Teste 2

A formulação matemática do problema teste 2 é similar à do problema teste 1, porém são utilizados os valores  $\alpha = 1$  e  $\theta = 10$ . A curva de Pareto nominal para o problema teste 2 é a mesma para o problema teste 1 e é apresentada na Fig. 8.1.

A Figura 8.3(a) apresenta a curva de Pareto nominal para o problema teste 2 e, para quatro valores do parâmetro de incerteza  $\delta$ , suas respectivas curvas de Pareto robustas analíticas, obtidas de maneira inteiramente análoga às curvas robustas para o problema teste 1. Nesta figura observa-se que o aumento da incerteza faz com que a amplitude de soluções não dominadas seja bem menor, caracterizando a redução do número de pontos viáveis na solução ótima de Pareto. A Figura 8.3(b) apresenta as curvas obtidas pelo MOFA para valores distintos do parâmetro de incerteza e mostra claramente o efeito do parâmetro  $\delta$  sobre o perfil das curvas de Pareto. Observa-se graficamente que, para cada valor de  $\delta$ , houve convergência do algoritmo MOFA para a solução analítica e também observa-se que as soluções estão bem distribuídas sobre seu perfil, não havendo concentração de soluções em uma região específica.

A Tabela 8.4 apresenta os resultados das métricas CM e DM para o problema teste

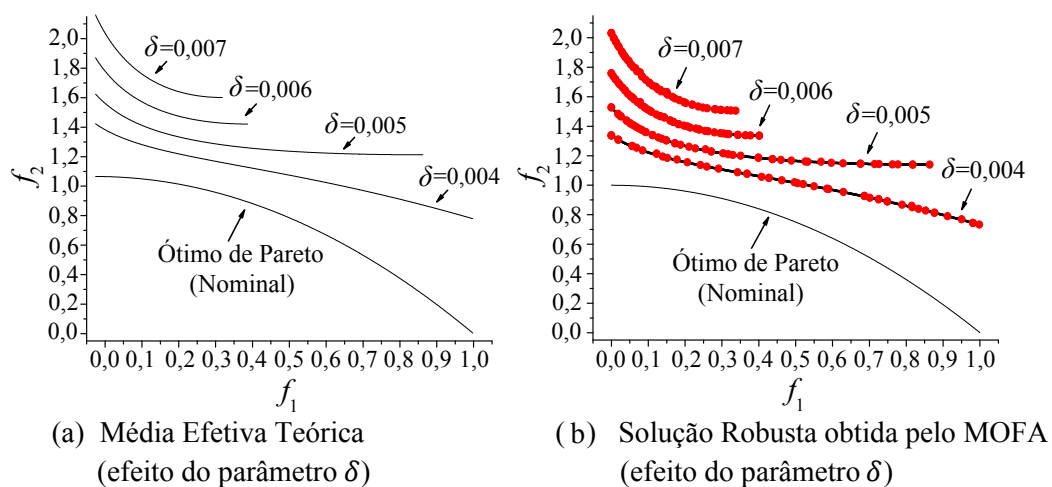


Figura 8.3: Curvas de Pareto robustas para o problema Teste 2.

2 robusto. A observação gráfica da convergência e da diversidade das soluções pode ser conferida analiticamente através dos resultados apresentados na Tab. 8.4. Em todos os casos houve convergência para a solução analítica robusta do problema teste 2.

Tabela 8.4: Resultados das métricas CM e DM na simulação do problema teste 2 robusto.

		<i>Simulações</i>									
$\delta$	<i>Métricas</i>	1	2	3	4	5	6	7	8	9	10
0,004	CM	0,001	0,001	0,001	0,002	0,002	0,002	0,003	0,002	0,002	0,002
	DM	0,849	0,816	0,808	1,038	0,847	0,889	0,814	0,823	0,863	0,982
0,005	CM	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001
	DM	0,755	0,753	0,670	0,928	0,978	0,755	0,859	0,678	0,989	0,788
0,006	CM	0,001	0,001	0,001	0,001	0,001	0,002	0,001	0,001	0,001	0,001
	DM	0,718	0,827	0,745	0,733	0,656	0,989	0,744	0,833	0,767	0,677
0,007	CM	0,001	0,001	0,001	0,002	0,001	0,001	0,001	0,001	0,001	0,001
	DM	0,726	0,744	0,879	0,989	0,656	0,767	0,894	0,799	0,989	0,748

### 8.3 Problema Teste 3

A formulação matemática do problema teste 3 é dada por (DEB;GUPTA, 2006):

$$\text{minimizar } \vec{F}_3(x) = (x_1, h(x_2) (g(x) + s(x_1))) \quad (8.11)$$

$$\begin{cases} h(x_2) = 2 - 0.8 \exp\left(-\left(\frac{x_2 - 0,35}{0,25}\right)^2\right) - \exp\left(-\left(\frac{x_2 - 0,85}{0,03}\right)^2\right) \\ g(x) = \sum_{i=3}^5 50x_i^2 \\ s(x_1) = 1 - \sqrt{x_1} \\ 0 \leq x_1, x_2 \leq 1, \quad -1 \leq x_i \leq 1 \quad i = 3, \dots, 5. \end{cases} \quad (8.12)$$

Este estudo de caso apresenta uma característica bastante interessante, qual seja a existência de uma curva de Pareto local. As curvas de Pareto global (ou somente a curva de Pareto) e local correspondem a  $x_i = 0$ ,  $i = 3, 4, \dots, 5$  o que faz com que  $g(x) = 0$ . Então, sobre as curvas,  $f_2(x_1, x_2) = h(x_2)s(x_1)$  e uma vez que  $f_1(x_1) = x_1$ , as Curvas de Pareto global e a local são obtidas respectivamente no mínimo global e local de  $h(x_2)$ . O mínimo global de  $h$  é obtido para  $x_2^* \approx 0,85$  (com  $h(x_2^*) \approx 1$ ) e o mínimo local de  $h$  é obtido em  $x_2^{**} = 0,35$  (com  $h(x_2^{**}) \approx 1,2$ ). Relações aproximadas entre os objetivos  $f_1$  e  $f_2$ , sobre as curvas global e local, são expressas por (DEB;GUPTA, 2006):

$$f_2 = 1 - \sqrt{f_1} \quad (\text{Global}) \quad (8.13)$$

$$f_2 = 1,2(1 - \sqrt{f_1}) \quad (\text{Local}). \quad (8.14)$$

Analogamente aos problemas testes 1 e 2, a curva de Pareto robusta, para este estudo de caso, pode ser obtida analiticamente mediante o cálculo das médias efetivas para os objetivos  $f_1$  e  $f_2$  em uma  $\delta$ -vizinhança do vetor de projeto  $x$ . Os objetivos obtidos pelas



médias efetivas, avaliadas sobre o conjunto de Pareto, são dadas pelas seguintes expressões:

$$f_1^{eff}(x, \delta) = x_1 \quad (8.15)$$

$$f_2^{eff}(x, \delta) = H(x_2^*, \delta_2) \left[ \sum_{i=3}^5 \frac{50}{3} \delta_i^2 + \left( 1 - \frac{1}{3\delta_i} \left( (x_1 + \delta_i)^{1,5} - (x_1 - \delta_i)^{1,5} \right) \right) \right] \quad (8.16)$$

onde

$$H(x_2^*, \delta_2) = \frac{1}{2\delta_2} \int_{x_2^* - \delta_2}^{x_2^* + \delta_2} h(y) dy. \quad (8.17)$$

A Figura 8.4 apresenta as curvas de Pareto global, local e a solução obtida pelo algoritmo MOFA, todas para o caso nominal. A existência da curva de Pareto local pode causar dificuldades de convergência para qualquer método de otimização, principalmente nos métodos determinísticos. A Fig. 8.4 mostra a convergência do algoritmo MOFA para a solução ótima global do problema teste 3. Observa-se, pelo menos graficamente, que as soluções estão bem espalhadas sobre a curva de Pareto.

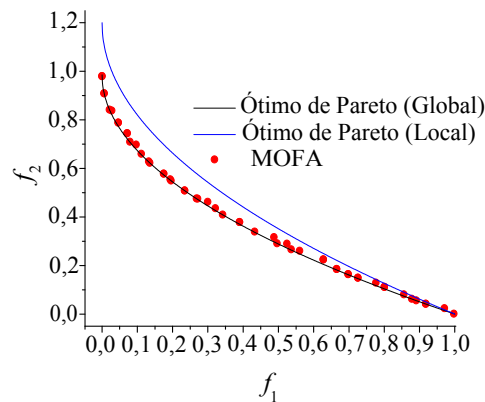


Figura 8.4: Curva de Pareto nominais (global e local) e obtida pelo MOFA para o problema teste 3.

A Tabela 8.5 mostra que, exceto na simulação com a semente inicial igual a zero,

ocorreu a convergência do algoritmo MOFA para a curva de Pareto global do problema teste 3. Através da Tabela 8.5 conclui-se também que as soluções estão bem distribuídas sobre a curva de Pareto global, isto é, há uma boa diversidade entre as soluções.

Tabela 8.5: Resultados das métricas CM e DM na simulação do problema teste 3 nominal.

<i>Simulações</i>										
<i>Métricas</i>	1	2	3	4	5	6	7	8	9	10
CM	0,056	0,009	0,014	0,013	0,021	0,007	0,020	0,009	0,015	0,012
DM	0,423	0,810	1,117	0,660	1,201	0,745	1,190	0,832	1,160	0,970
<i>Ótimo</i>	Local	Global	Global	Global	Global	Global	Global	Global	Global	Global

Para estudar o efeito do parâmetro  $\delta$  sobre a solução (curva de Pareto) foi utilizado o seguinte vetor de perturbação  $[\delta_1, \delta_2, \delta_3, \delta_4, \delta_5] = [\delta \ \delta \ 2\delta \ 2\delta \ 2\delta]$ , atuando sobre o vetor de projeto  $x$  (DEB;GUPTA, 2006). A Figura 8.5(a) apresenta as curvas de Pareto global e local robustas, obtidas analiticamente através das equações 8.15 e 8.16, para três valores do parâmetro de perturbação  $\delta$ . A Figura 8.5(b) apresenta as curvas de Pareto robustas obtidas pelo algoritmo MOFA para o problema teste 3. Observa-se que há convergência para a curva solução global para os três valores de  $\delta$  selecionados. Observa-se também que as soluções estão bem distribuídas sobre as respectivas curvas de Pareto robustas.

Através da Tabela 8.6 conclui-se que em todas simulações houve convergência do algoritmo para a solução ótima global do problema teste 3. Os valores da métrica DM indicam que as soluções realmente estão bem distribuídas sobre as curvas de Pareto robustas.

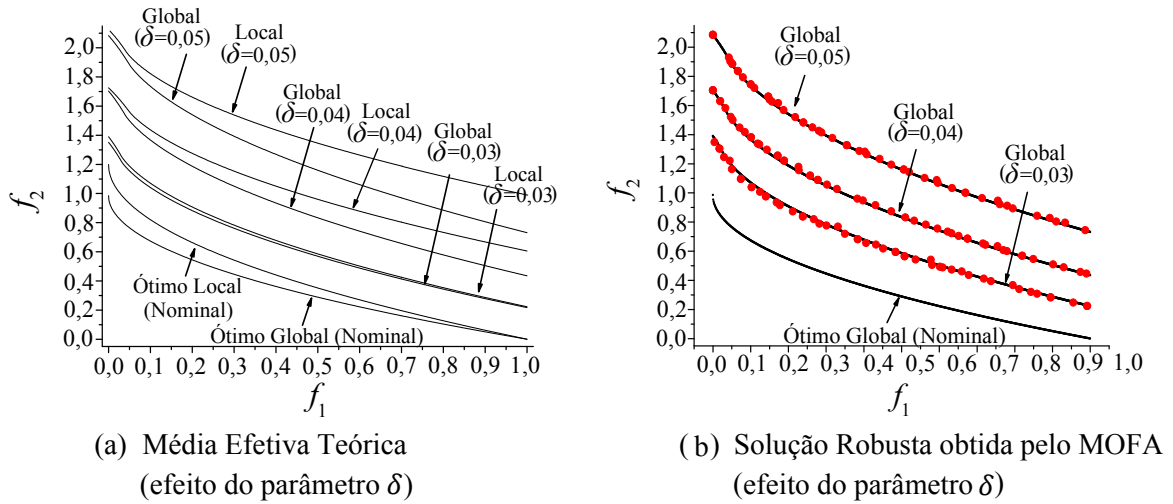


Figura 8.5: Curvas de Pareto robustas para o problema teste 3.

Tabela 8.6: Resultados das métricas CM e DM na simulação do problema teste 3 robusto.

		<i>Simulações</i>									
$\delta$	<i>Métricas</i>	1	2	3	4	5	6	7	8	9	10
0,03	CM	0,007	0,006	0,006	0,005	0,006	0,006	0,004	0,007	0,005	0,005
	DM	0,772	0,819	0,946	0,901	1,062	0,838	1,329	0,703	0,793	0,903
<i>Ótimo</i>		Global	Global	Global	Global	Global	Global	Global	Global	Global	Global
0,04	CM	0,005	0,004	0,006	0,003	0,005	0,004	0,001	0,002	0,004	0,003
	DM	0,905	0,765	0,888	0,876	0,874	0,915	0,875	0,985	0,577	0,751
<i>Ótimo</i>		Global	Global	Global	Global	Global	Global	Global	Global	Global	Global
0,05	CM	0,003	0,005	0,004	0,005	0,005	0,004	0,005	0,004	0,005	0,003
	DM	0,878	0,951	0,885	0,951	0,889	0,844	0,781	0,998	0,778	0,985
<i>Ótimo</i>		Global	Global	Global	Global	Global	Global	Global	Global	Global	Global

## 8.4 Problema Teste 4

A formulação matemática para o problema teste 4 é dada por:

$$\text{minimizar } \vec{F}_4(x) = (x_1, h(x_1) + g(x)s(x_1)) \quad (8.18)$$

$$\text{sujeito a } r(x) = 0, 2x_1 + x_2 - 0, 1 \geq 0 \quad (8.19)$$

$$\begin{cases} h(x_1) = 1 - x_1^2, & g(x) = \sum_{i=2}^5 50x_i^2 \\ s(x_1) = \frac{1}{0, 2 + x_1} + x_1^2 \\ 0 \leq x_1 \leq 1, & -1 \leq x_i \leq 1 \quad i = 2, \dots, 5. \end{cases} \quad (8.20)$$

Isolando a variável  $x_2$  na inequação, que define a restrição, observa-se que a restrição é sempre atendida para valores de  $x_1 \geq 0,5$ . A curva de Pareto nominal restrita é obtida ativando a restrição, isto é, fazendo  $x_2 = 0,1 - 0,2x_1$  e considerando  $x_1 \in [0,1]$  e  $x_i = 0$ ,  $i = 3, \dots, 5$ . Assim, a relação entre os objetivos  $f_1$  e  $f_2$ , para o problema nominal restrito, é dada por (DEB;GUPTA, 2006):

$$f_2 = \begin{cases} 1 - f_1^2 & \text{se } f_1 \geq 0,5 \\ 1 - f_1^2 + 50(0,1 - 0,2f_1)^2 \left( \frac{1}{0,2 + f_1} + f_1^2 \right) & \text{se } f_1 < 0,5 \end{cases} \quad (8.21)$$

De forma inteiramente análoga aos casos anteriores, onde as médias efetivas são calculadas analiticamente a partir da definição, os objetivos obtidos pelas médias efetivas, avaliados sobre o conjunto de Pareto, são expressos pelas seguintes funções:

$$f_1^{eff}(x, \delta) = x_1 \quad (8.22)$$

$$f_2^{eff}(x, \delta) = \begin{cases} 1 - x_1^2, & x_1 \geq 0,5 \\ H_1 H_2, & x_1 < 0,5. \end{cases} \quad (8.23)$$

onde

$$H_1 = 1 - x_1^2 - \frac{\delta_1^2}{3} + \left[ \frac{1}{2\delta_1} \ln \left( \frac{0,2 + x_1 + \delta_1}{0,2 + x_1 - \delta_1} \right) + \left( x_1^2 + \frac{\delta_1^2}{3} \right) \right] \sum_{i=3}^5 \frac{100}{3} \delta_i^2 \quad (8.24)$$

e

$$H_2 = \frac{50}{2\delta_1} \int_{x_1 - \delta_1}^{x_1 + \delta_1} \left( \frac{1}{0,2 + y} + y^2 \right) (0,1 - 0,2y + 0,2\delta_1 + \delta_2)^2 dy \quad (8.25)$$

A Figura 8.6 apresenta as curvas de Pareto nominais sem e com a restrição e também

apresenta a solução obtida pelo MOFA para o caso nominal restrito. Pela figura observa-se que há convergência do algoritmo MOFA para a curva de Pareto restrita e que há também uma boa distribuição das soluções sobre o perfil da curva solução, indicando assim um bom desempenho do algoritmo para o caso nominal restrito. Tanto para o caso nominal quanto para o robusto foi utilizado o fator de penalidade  $r_p$ , para tratamento da restrição, com valor constante e igual a  $10^6$ .

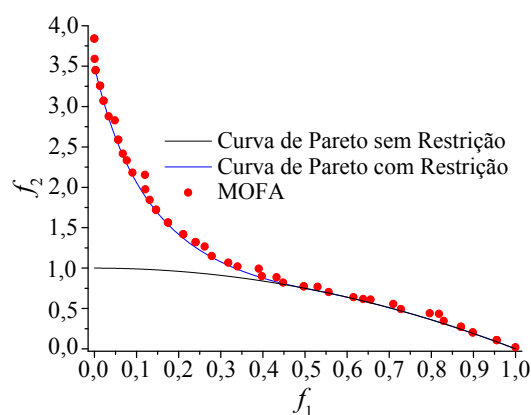


Figura 8.6: Curva de Pareto nominal e obtida pelo algoritmo MOFA para o problema teste 4.

A Tabela 8.7 apresenta os resultados das métricas de convergência (CM) e de diversidade (DM) para o problema teste 4 nominal. A observação gráfica da convergência e da diversidade das soluções, apresentada na Fig. 8.6, podem ser comprovadas analiticamente pelos resultados das métricas CM e DM apresentados na Tab. 8.7.

Tabela 8.7: Resultados das métricas CM e DM na simulação do problema teste 4 nominal.

Simulações										
Métricas	1	2	3	4	5	6	7	8	9	10
CM	0,0255	0,0312	0,0193	0,0215	0,0194	0,0383	0,0222	0,0372	0,0254	0,0166
DM	0,9748	1,0634	0,6557	0,6258	0,4846	1,2954	0,7239	1,0779	0,9115	0,6725

Para estudar o efeito do parâmetro de incerteza sobre a curva de Pareto, foram utilizados os seguintes valores  $\delta = [0,010 \ 0,015 \ 0,020]$  e o vetor de perturbação das cinco

variáveis de projeto foi definido por  $[\delta_1, \delta_2, \delta_3, \delta_4, \delta_5] = [\delta \ 2\delta \ 2\delta \ 2\delta \ 2\delta]$ . Na Fig. 8.7 foram apresentados os resultados obtidos pelo algoritmo MOFA juntamente com as curvas de Pareto analíticas robustas para os três valores do parâmetro de incerteza  $\delta$ . Nos três casos houve convergência das soluções obtidas pelo MOFA para a curva solução analítica do problema. Observa-se também uma boa distribuição das soluções sobre as curvas. Conforme destacado por Deb e Gupta (2006), a curva de Pareto analítica pode ser descrita por duas regiões, dentre as quais o efeito da restrição, em relação ao problema sem restrição, é evidenciado para  $x_1$  menor que 0,5. Por outro lado, para  $x_1$  maior que 0,5, a restrição é sempre atendida, independentemente do valor de  $\delta$  empregado. Este efeito fica claro com o aumento do parâmetro de perturbação  $\delta$  na Fig. 8.7, fazendo com que exista descontinuidade na curva de Pareto.

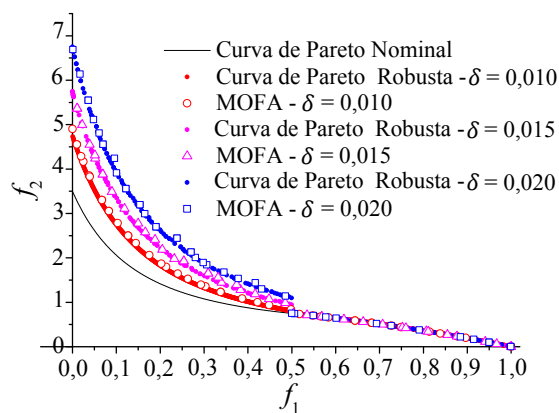


Figura 8.7: Curvas de Pareto robustas para o problema teste 4.

Para finalizar o estudo do problema teste 4, apresenta-se a seguir a Tab. 8.8 com os resultados das métricas de convergência (CM) e de diversidade (DM) obtidos nas simulações do algoritmo MOFA para os três valores do parâmetro de incerteza  $\delta$ . Nesta tabela observa-se que os valores obtidos para as métricas são considerados satisfatórios, já que os valores da métrica de convergência tendem a valores pequenos e os valores da métrica de diversidade, juntamente com a distribuição das soluções mostrada na Fig. 8.7,

mostram que há uma boa diversidade das soluções.

Tabela 8.8: Resultados das métricas CM e DM na simulação do problema teste 4 robusto.

$\delta$	Métricas	Simulações									
		1	2	3	4	5	6	7	8	9	10
0,010	CM	0,1611	0,1205	0,1623	0,0673	0,0395	0,0807	0,0924	0,0433	0,0567	0,1670
	DM	1,3729	1,3669	1,4519	1,0982	0,7061	1,0574	1,1956	0,7791	1,0347	1,4293
0,015	CM	0,1451	0,1154	0,1435	0,0445	0,0454	0,0798	0,0788	0,0409	0,0457	0,1799
	DM	1,3095	1,2789	1,4098	1,0244	0,6787	1,0877	1,0126	0,7567	1,0336	1,4098
0,020	CM	0,1141	0,1055	0,1243	0,0563	0,0309	0,0909	0,0894	0,0435	0,0709	0,1454
	DM	1,9555	1,2329	1,5339	1,0878	0,7121	1,0677	1,3426	0,7666	1,0333	1,3898

## 8.5 Problema Teste 5

Dentre todas as funções-teste apresentadas na literatura, as funções ZDT (Zitzler-Deb-Thiele) desenvolvidas por Zitzler et al. (2000) são as mais utilizadas para validação de algoritmos multiobjetivos. Isto se deve à dificuldade de convergência encontrada pelos algoritmos. Estas dificuldades estão associadas à convexidade e não convexidade, fronteiras discretas e ou descontínuas e multimodalidade (LOBATO, 2008).

Escolheu-se uma das funções ZDT objetivando o estudo da convergência do algoritmo MOFA, da diversidade das soluções obtidas e também para verificação do comportamento das soluções robustas em função do parâmetro de incerteza  $\delta$ . O problema de otimização que é estudado nesta seção é formulado matematicamente através das seguintes expressões (ZITZLER et al., 2000):

$$\text{minimizar } \vec{F}_5(x) = (x_1, g(x_2, \dots, x_5)h(x_1, g(x_2, \dots, x_5))) \quad (8.26)$$

$$\begin{cases} g(x_2, \dots, x_5) = 1 - \frac{9}{5-1} \sum_{i=2}^5 x_i, \\ h(x) = 1 - \sqrt{\frac{x_1}{g(x_2, \dots, x_5)}} - \frac{x_1}{g(x_2, \dots, x_5)} \sin(10\pi x_1), \\ 0 < x_i < 1, \quad i = 1, \dots, 5. \end{cases} \quad (8.27)$$

Comprovando os resultados de Lobato (2008), o problema teste 5 possui curva de Pareto descontínua provocada pela presença da função *seno* em  $h$ . A curva de Pareto ocorre sobre a curva determinada por  $x_1 \in (0, 1)$  e  $x_i = 0$ ,  $i = 2, \dots, 5$ . Houve apenas uma mudança nos parâmetros do algoritmo MOFA presentes na Tab. 8.1; foram necessárias 500 gerações.

A Figura 8.8 apresenta a curva fronteira do espaço de objetivos, a curva de Pareto nominal do problema 8.26 e a curva solução nominal obtida pelo MOFA. A descontinuidade dificulta a convergência do algoritmo e também uma boa distribuição das soluções, porém, observa-se a convergência do algoritmo para a solução nominal, em todos os casos, e que há também uma boa diversidade das soluções sobre a curva de Pareto.

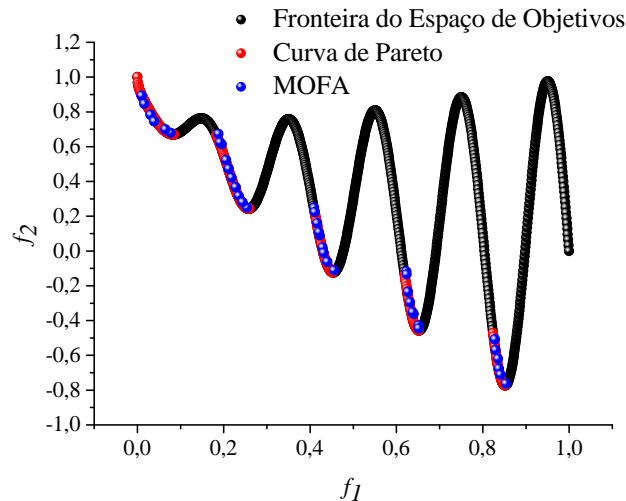


Figura 8.8: Curva de Pareto nominal e obtida pelo algoritmo MOFA para o problema teste 5.



A Tabela 8.9 apresenta os resultados das métricas de convergência CM e de diversidade DM para as dez simulações, usando o vetor de sementes iniciais citado anteriormente. Em todas simulações do algoritmo MOFA houve convergência para a curva de Pareto nominal e o algoritmo manteve boa diversidade das soluções sobre a curva de Pareto.

Tabela 8.9: Resultados das métricas CM e DM na simulação do problema teste 5 nominal.

<i>Simulações</i>										
<i>Métricas</i>	1	2	3	4	5	6	7	8	9	10
CM	0,0052	0,0085	0,0041	0,0084	0,0061	0,0068	0,0049	0,0057	0,0094	0,0062
DM	0,8655	0,7688	0,8608	0,8044	0,8648	0,8090	0,8649	0,8417	0,7570	0,8507

A Figura 8.9 apresenta a curva solução analítica nominal, a curva de Pareto nominal e duas curvas robustas para os valores  $\delta = 0,01$  e  $\delta = 0,02$ . Foi utilizado o seguinte vetor de parâmetros  $[\delta_1 \delta_2 \delta_3 \delta_4 \delta_5] = [\delta \delta \delta \delta \delta]$ . Observa-se o efeito do parâmetro  $\delta$  nas soluções robustas em relação à curva nominal, isto é, sem robustez. Observa-se que as soluções robustas estão bem distribuídas em cada parte que compõe a curva nominal e portanto, para o caso robusto, o algoritmo MOFA assegurou a diversidade entre as soluções.

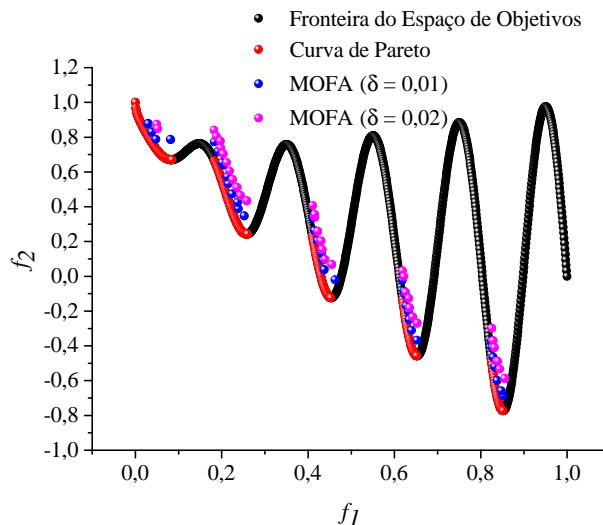


Figura 8.9: Curvas de Pareto robustas para o problema teste 5.

## 8.6 Problema Teste 6

O problema teste 6, estudado nesta seção, foi proposto por Fonseca e Fleming (1995).

Sua formulação matemática é dada por:

$$\text{minimizar } \vec{F}_6(x) = (f_1(x_1, \dots, x_5), f_2(x_1, \dots, x_5)) \quad (8.28)$$

$$\begin{cases} f_1(x_1, \dots, x_5) = 1 - \exp\left(-\sum_{i=1}^5 \left(x_i - \frac{1}{\sqrt{5}}\right)^2\right), \\ f_2(x_1, \dots, x_5) = 1 - \exp\left(-\sum_{i=1}^5 \left(x_i + \frac{1}{\sqrt{5}}\right)^2\right), \\ -4 < x_i < 4, \quad i = 1, \dots, 5. \end{cases} \quad (8.29)$$

Este problema é altamente influenciado pela função exponencial nos objetivos. A curva de Pareto é a imagem dos objetivos avaliada sobre o conjunto  $x_i \in (-1/\sqrt{5}, 1/\sqrt{5})$ ,  $i = 1, \dots, 5$ . A Figura 8.10 apresenta a curva de Pareto e o conjunto de soluções obtido pelo algoritmo MOFA. Os parâmetros do algoritmo para as simulações nesta seção são

idênticos aos da Tab. 8.1. Observa-se a convergência e uma boa distribuição das soluções obtidas pelo MOFA sobre a curva de Pareto para o caso nominal.

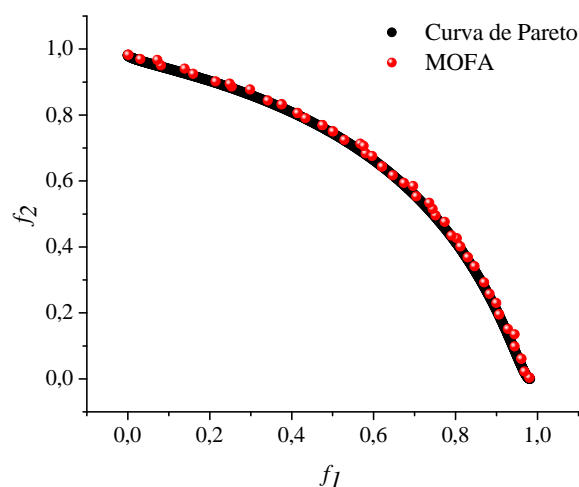


Figura 8.10: Curva de Pareto nominal e obtida pelo algoritmo MOFA para o problema teste 6.

A Tabela 8.10 apresenta os resultados das métricas CM e DM obtidos pelo MOFA para o caso nominal. Observa-se, em todos os casos, convergência do algoritmo e uma boa distribuição das soluções sobre a curva de Pareto. A função exponencial, decrescente neste caso, faz com que os objetivos diminuam rapidamente, garantindo, assim, que a convergência seja rápida. Por outro lado, se não houvesse um mecanismo de diversidade eficiente, a tendência é que as soluções se aglomerassem na região central da curva de Pareto. Portanto o mecanismo de diversidade do MOFA se mostrou eficiente neste caso.

Tabela 8.10: Resultados das métricas CM e DM na simulação do problema teste 6 nominal.

<i>Simulações</i>										
<i>Métricas</i>	1	2	3	4	5	6	7	8	9	10
CM	0,0065	0,0059	0,0062	0,0070	0,0056	0,0064	0,0051	0,0067	0,0061	0,0061
DM	0,7559	0,6665	0,5961	0,6983	0,7463	0,6955	0,6914	0,5915	0,6737	0,7569

Das definições de média efetiva e do problema teste 6 não é possível obter analiticamente os objetivos robustos, dados pelas médias efetivas dos objetivos nominais. Assim não será possível a comparação entre as soluções robustas obtidas pelo MOFA e a curva de Pareto robusta. A análise da robustez será apenas para comparar as soluções robustas, em função do parâmetro de incerteza  $\delta$ , com a solução nominal.

Devido ao comportamento da função exponencial, valores do parâmetro de incerteza  $\delta \leq 0.1$  não são suficientes para diferenciar entre a curva nominal e a analítica. Para perturbar a curva solução, foi necessário utilizar valores considerados grandes para o parâmetro de incerteza  $\delta$ .

A Figura 8.11 apresenta a curva de Pareto nominal e algumas curvas robustas obtidas pelo MOFA considerando  $\delta = [0, 1 \ 0, 3 \ 0, 5 \ 0, 7]$ . O seguinte vetor de parâmetros  $[\delta_1 \ \delta_2 \ \delta_3 \ \delta_4 \ \delta_5] = [\delta \ \delta \ \delta \ \delta \ \delta]$  foi utilizado nas simulações e os outros parâmetros do algoritmo MOFA são os mesmos que estão contidos na Tab. 8.1. Observa-se o efeito do crescimento do parâmetro  $\delta$  no perfil das curvas robustas, mostrando que o aumento da robustez gera mudanças significativas sobre as soluções robustas.

## 8.7 Problema Teste 7

O problema teste 7, estudado nesta seção, é uma pequena modificação do problema estudado na seção anterior. Sua formulação matemática é dada por:

$$\text{minimizar } \vec{F}_7(x) = (f_1(x_1, \dots, x_5), f_2(x_1, \dots, x_5)) \quad (8.30)$$

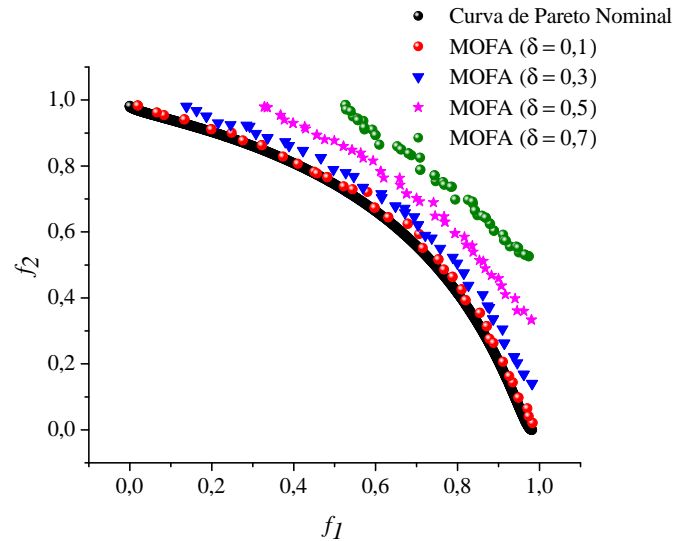


Figura 8.11: Curvas de Pareto robustas para o problema teste 6.

$$\begin{cases} f_1(x_1, \dots, x_5) = 1 - \exp\left(\sum_{i=1}^5 \left(x_i - \frac{1}{\sqrt{5}}\right)^2\right), \\ f_2(x_1, \dots, x_5) = 1 - \exp\left(\sum_{i=1}^5 \left(x_i + \frac{1}{\sqrt{5}}\right)^2\right), \\ -0,1 < x_i < 0,1 \quad i = 1, \dots, 5. \end{cases} \quad (8.31)$$

As únicas alterações com relação ao problema anterior, definido pelas equações 8.28 e 8.29, estão nas restrições laterais (das variáveis de projeto) e no sinal no interior das funções exponenciais nos objetivos, negativos no caso anterior e positivos no caso atual. A curva de Pareto analítica nominal não é conhecida. Sendo assim, não será possível realizar um estudo da convergência e da diversidade das soluções neste estudo de caso. O estudo realizado é dedicado à comparação entre a curva nominal, obtida pelo MOFA sem robustez, com as soluções robustas, também obtidas pelo MOFA, para alguns valores do parâmetro de incerteza  $\delta$ .

A Figura 8.12 apresenta a curva solução obtida pelo MOFA sem robustez e três curvas robustas obtidas pelo algoritmo para  $\delta = [0,05 \ 0,07 \ 0,09]$ . Os parâmetros do algoritmo e

do vetor de perturbação, usados nesta seção, são os mesmos que foram utilizados na seção anterior. Observa-se que, com o aumento do parâmetro  $\delta$ , parece haver uma pequena rotação no sentido horário das curvas robustas em relação à nominal. Neste caso fica evidenciado o efeito do parâmetro de robustez  $\delta$  no perfil das soluções robustas.

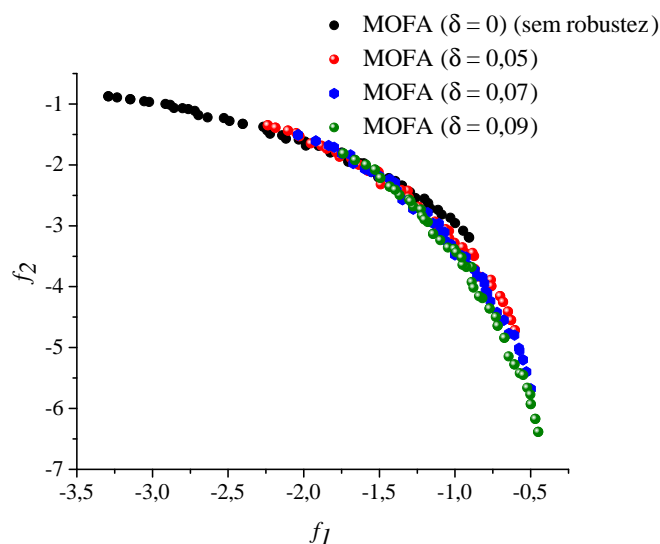


Figura 8.12: Curvas de Pareto robustas para o problema teste 7.

## 8.8 Análise de Sensibilidade dos Parâmetros do MOFA

Nesta seção será realizada a análise da sensibilidade dos seguintes parâmetros do algoritmo MOFA: coeficiente de atratividade máxima  $\beta_0$ , coeficiente de absorção da luz  $\gamma$  e do número de pontos amostrados para estimar a integral  $N_{amt}$ . O problema estudado é o problema teste 1, da seção 8.1 deste capítulo e foi usado o valor  $\delta = 0,007$  para o parâmetro de perturbação. Foram analisadas as respostas referentes às métricas de convergência (CM) e de diversidade (DM). Para cada um dos casos estudados, foram realizadas 10 simulações.

Yang (2008), proponente do algoritmo ACV, propõe que os parâmetros  $\beta_0$  e  $\gamma$  variem nos intervalos  $[0, 1]$  e  $[0, 10]$  respectivamente. Sabe-se que o parâmetro  $N_{\text{amt}}$ , usado para estimar a integral da definição de média efetiva, deve ser tomado de forma que a estimativa forneça uma boa aproximação da integral, porém deve ser tomado de tal forma que não aumente muito a quantidade de cálculos realizados pelo algoritmo MOFA.

Para analisar a sensibilidade dos parâmetros do algoritmo MOFA, foram escolhidos os seguintes valores para a variação dos parâmetros de estudo:

$$\beta_0 = [0 \ 0,1 \ 0,2 \ 0,3 \ 0,4 \ 0,5 \ 0,6 \ 0,7 \ 0,8 \ 0,9 \ 1,0]; \quad (8.32)$$

$$\gamma = [0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10]; \quad (8.33)$$

$$N_{\text{amt}} = [2 \ 5 \ 20 \ 50 \ 100]. \quad (8.34)$$

### Convergência Métrica

Inicialmente serão analisadas as respostas do algoritmo MOFA referente à métrica de convergência, sujeitas a variação dos parâmetros considerados.

A Figura 8.13 mostra os resultados obtidos considerando os seguintes parâmetros fixos  $\gamma = 1$ ,  $\delta = 0,007$ ,  $N_{\text{vag}} = 55$ ,  $N_{\text{ger}} = 250$  e variando o parâmetro  $\beta_0$ .

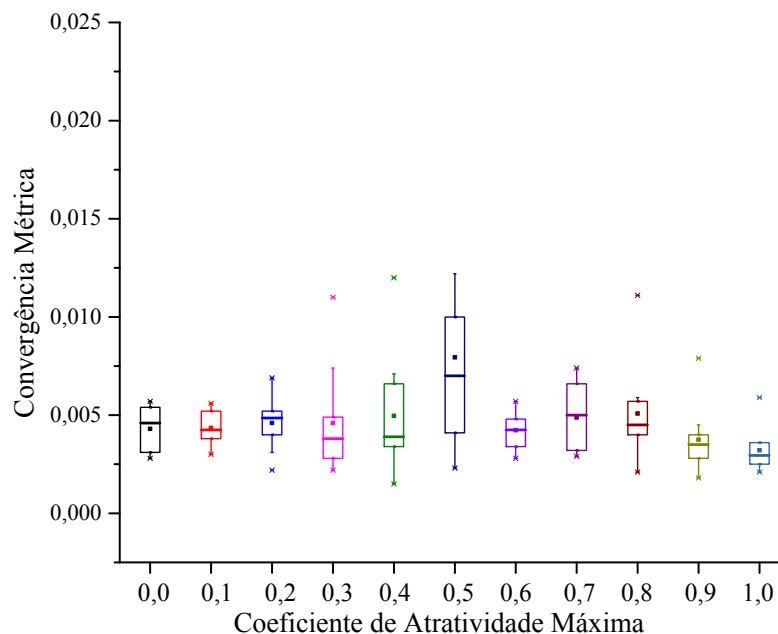


Figura 8.13: Sensibilidade do algoritmo MOFA. Métrica de convergência *versus* variação de  $\beta_0$ .

Para cada valor do parâmetro  $\beta_0$ , a caixa retangular, apresentada na figura anterior, representa os valores entre o primeiro e o terceiro quartil, a linha mais espessa, interior a cada caixa retangular, representa a mediana dos dados, o ponto retangular, também interior à caixa, representa a média aritmética dos dados e os pequenos símbolos x representam o maior e o menor valor obtidos nas simulações. Observa-se que em todos os casos houve a convergência do algoritmo MOFA. Observou-se pequena sensibilidade do algoritmo em relação ao coeficiente  $\beta_0$  e que o melhor valor para a métrica CM foi obtido quando foi utilizado  $\beta_0 = 1$  e que, para este valor do parâmetro  $\beta_0$ , a variação dos valores da métrica de convergência CM foi menor do que para outros parâmetros.

A Figura 8.14 apresenta os resultados da métrica CM sujeitos à variação do coeficiente de absorção da luz  $\gamma$ , considerando o parâmetro  $\beta_0 = 1$  e os outros dados da simulação



são idênticos aos da análise anterior.

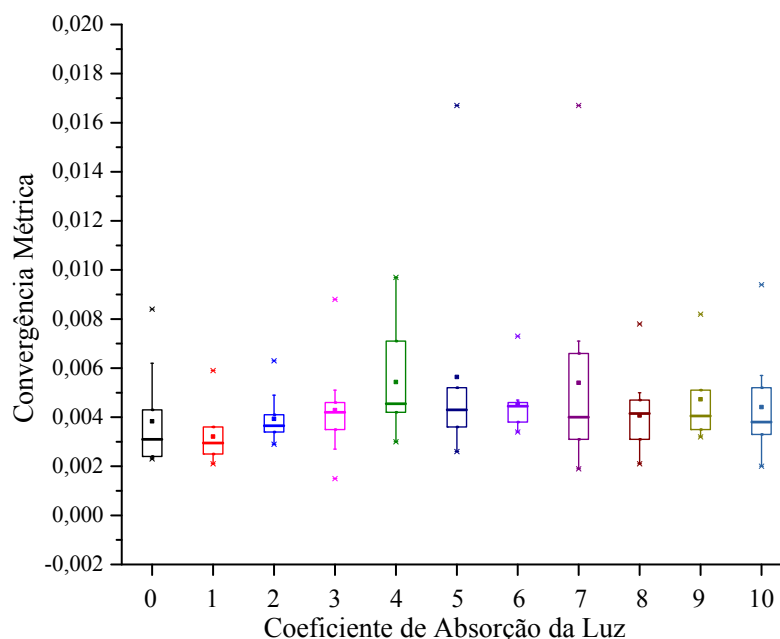


Figura 8.14: Sensibilidade do algoritmo MOFA. Métrica de convergência *versus* variação de  $\gamma$ .

A Figura 8.14 mostra que os resultados da métrica CM foram melhores quando  $\gamma = 1$ . Observa-se que a variação dos dados, quando  $\gamma = 1$ , apesar de não ser a menor, foi pequena. Assim o melhor valor, para este estudo de caso, para o coeficiente de absorção da luz é  $\gamma = 1$ .

Por último, a Figura 8.15 apresenta os resultados da métrica CM variando os valores para o tamanho da amostra escolhida para estimar a integral da definição de média efetiva.

O valor mínimo para o tamanho da amostra é  $N_{\text{amt}} = 2$ . Observa-se que para  $N_{\text{amt}} = 20$ , houve simulações para as quais o valor da métrica CM foi bastante pequeno, porém o valor médio é tão próximo de zero e a variação também não foi pequena. Observou-se que a partir de  $N_{\text{amt}} = 50$  os valores da métrica CM, obtidos nas simulações, é bastante pequeno, além da variação dos valores de CM serem bastante pequenos também. Sendo

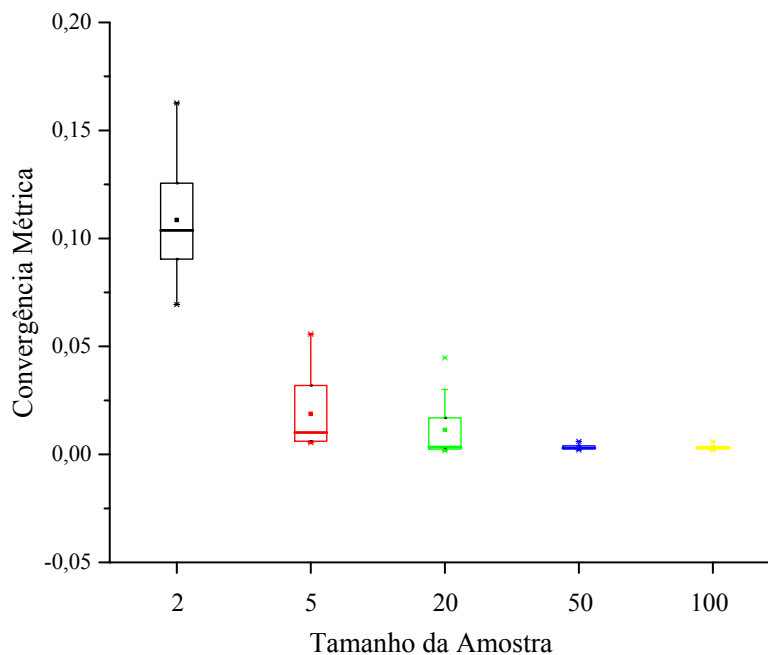


Figura 8.15: Sensibilidade do algoritmo MOFA. Métrica de convergência *versus* variação  $N_{amt}$ .

assim, observou-se, para este estudo de caso, que a partir de  $N_{amt} = 50$  a estimativa da integral foi eficiente na tarefa de aproximar o valor exato da média efetiva.

#### Diversidade Métrica

Neste momento serão analisadas as respostas do algoritmo MOFA referente à métrica de diversidade, sujeitas a variação dos parâmetros considerados.

A Figura 8.16 mostra os resultados obtidos considerando os seguintes parâmetros fixos  $\gamma = 1$ ,  $\delta = 0,007$ ,  $N_{vag} = 55$ ,  $N_{ger} = 250$  e variando o parâmetro  $\beta_0$ .

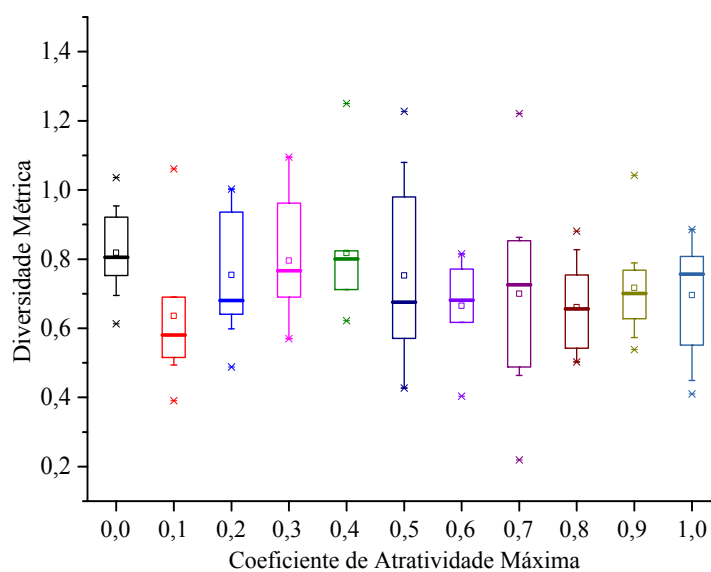


Figura 8.16: Sensibilidade do algoritmo MOFA. Métrica de diversidade *versus* variação de  $\beta_0$ .

A Figura 8.17 mostra os resultados obtidos considerando os seguintes parâmetros fixos  $\beta_0 = 1$ ,  $\delta = 0,007$ ,  $N_{\text{vag}} = 55$ ,  $N_{\text{ger}} = 250$  e variando o parâmetro  $\gamma$ .

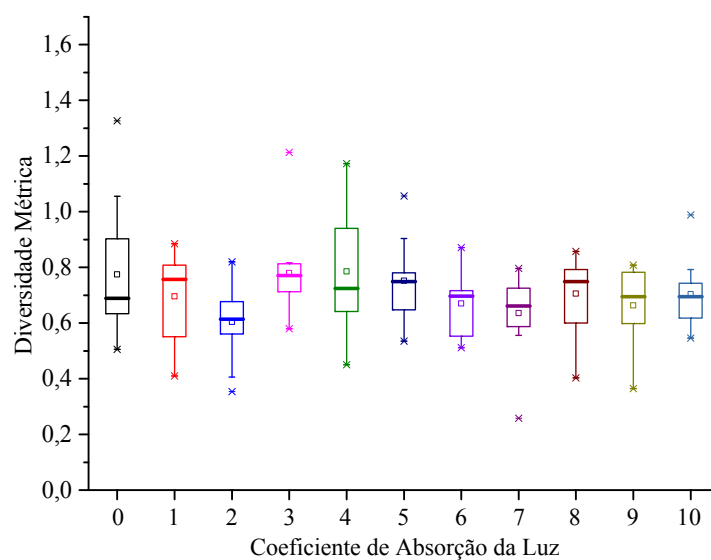


Figura 8.17: Sensibilidade do algoritmo MOFA. Métrica de diversidade *versus* variação de  $\gamma$ .

A Figura 8.18 mostra os resultados obtidos considerando os seguintes parâmetros fixos  $\beta_0 = \gamma = 1$ ,  $\delta = 0,007$ ,  $N_{\text{vag}} = 55$ ,  $N_{\text{ger}} = 250$  e variando o parâmetro  $N_{\text{amt}}$ .

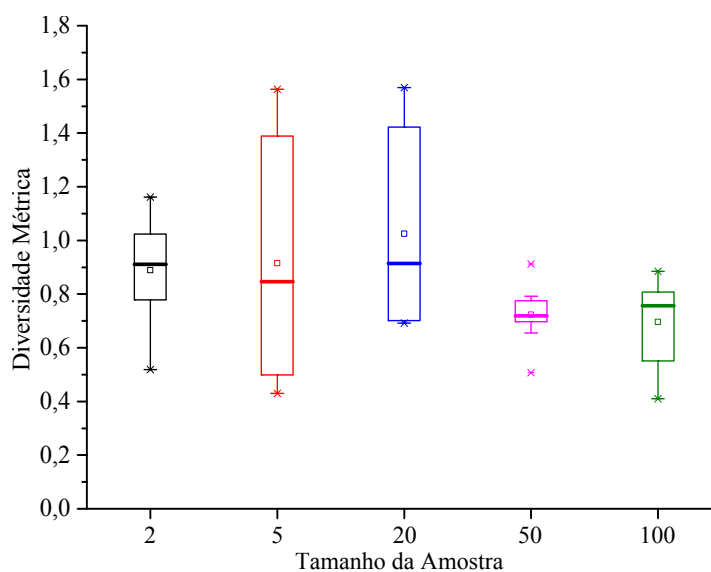


Figura 8.18: Sensibilidade do algoritmo MOFA. Métrica de diversidade *versus* variação de  $N_{amt}$ .

As Figuras 8.16, 8.17 e 8.18 apresentam os resultados da métrica de diversidade, obtidos pelo algoritmo MOFA a partir da variação dos parâmetros considerados. Pode-se perceber que, no geral, o algoritmo cumpriu eficazmente a tarefa de diversidade das soluções, ou seja, garantiu um bom espaçamento das soluções obtidas sobre a curva de Pareto.

## 8.9 Conclusões Preliminares

Nos problemas testes 1, 2, 3 e 4 tanto a Curva de Pareto nominal quanto a robusta já eram conhecidas. Sendo assim houve a possibilidade de avaliação da convergência do algoritmo MOFA para os casos nominal e robusto. Observou-se que nestes quatro estudos de caso o algoritmo MOFA convergiu para as soluções analíticas nominais e robustas. A avaliação da convergência foi obtida através dos valores da métrica de convergência CM, definida anteriormente neste capítulo pela Eq. 8.1. Além da convergência, uma das

---

metas específicas para problemas multiobjetivo é a diversidade entre as soluções, isto é, a distribuição das soluções. Observou-se também que o algoritmo MOFA foi eficiente ao distribuir as soluções e esta eficiência pôde ser comprovada tanto geometricamente, isto é, nas figuras deste capítulo, quanto analiticamente, através dos valores da métrica de diversidade DM, definida previamente pela Eq. 8.2.

Os problemas teste 1 e 2 apresentam multimodalidade devido à função *cos seno* em sua formulação. A multimodalidade pode dificultar a convergência do algoritmo, porém, como os valores da métrica de convergência comprovam, em todos os casos o algoritmo convergiu para as curvas de Pareto nominais e robustas. Observa-se também, em todas as simulações, uma boa distribuição das soluções sobre a Fronteira de Pareto.

No Problema Teste 3 aparece a figura da Fronteira de Pareto Local, para os casos nominal e robusto. Um algoritmo que utiliza alguma técnica determinística para convergência certamente enfrentaria dificuldades de convergência neste caso. Porém, em todas as simulações, apenas em um único caso o algoritmo MOFA convergiu para a Fronteira local.

O Problema Teste 4 é um problema com restrição. A técnica utilizada para tratar as restrições foi a de penalizar quando as soluções perturbadas violassem alguma das restrições. O número de soluções vizinhas que violam alguma das restrições é contado e o número é multiplicado por um fator de penalidade constante  $e$ , depois, o resultado é adicionado aos objetivos. Sendo assim, as soluções que estão na fronteira do espaço de projetos, mesmo que ótimas, são preteridas em relação àquelas soluções, mesmo que sub-ótimas, que possuem alguma distância da fronteira do espaço de projetos. O algoritmo MOFA convergiu, em todos os casos, para a Curva de Pareto Nominal e Robusta para o problema restrito.

O Problema Teste 5 possui Curva de Pareto descontínua. Neste caso, a Curva de Pareto robusta analítica não é conhecida. Além da convergência, a descontinuidade da Curva

---

de Pareto dificulta muito uma boa distribuição das soluções. Observou-se a convergência do algoritmo MOFA para a Fronteira de Pareto Nominal e também uma boa distribuição das soluções sobre a Curva de Pareto. Para o caso robusto, foram apresentadas duas curvas robustas mostrando o efeito do parâmetro de incerteza  $\delta$  no perfil das soluções. Sendo assim, o algoritmo MOFA mostrou-se eficiente durante a tarefa de distribuição das soluções, mantendo boa diversidade entre as mesmas.

O Problema Teste 6 também apresenta muitas dificuldades para que um algoritmo possa distribuir as soluções sobre a Curva de Pareto. Esta dificuldade é devido ao comportamento da função exponencial e das funções quadráticas que aparecem como argumento da função exponencial. Os objetivos decrescem muito rapidamente, fazendo com que as soluções se concentrem na região central da Curva de Pareto. O mecanismo de diversidade do algoritmo MOFA funcionou de forma satisfatória na tarefa de distribuir as soluções sobre a Fronteira de Pareto. Ainda devido ao comportamento da função exponencial, presente nos objetivos, pequenos valores do parâmetro de incerteza  $\delta$  não diferenciam as soluções robustas das nominais. Foi necessário utilizar valores considerados grandes para o parâmetro  $\delta$  para perceber sua influência no perfil das soluções robustas.

O Problema Teste 7 é similar ao problema teste 6. A Curva de Pareto Nominal não é conhecida. A Figura 8.12 apresentou as curvas sem robustez e robustas, para três valores de  $\delta$ , obtidas pelo MOFA. Nela é possível observar o efeito do parâmetro  $\delta$  na inserção da robustez ao problema.

De forma geral pode-se concluir que o algoritmo MOFA mostrou-se eficiente, tanto na convergência quanto na distribuição das soluções sobre a Curva de Pareto, pois os problemas estudados apresentam sérias dificuldades para a realização das tarefas de convergência e de diversidade e, no entanto, o algoritmo MOFA obteve êxito, para os problemas estudados, nas tarefas de convergência e diversidade das soluções.

A análise de sensibilidade mostrou, para o caso estudado, que os melhores valores para os parâmetros coeficiente de atratividade máxima, coeficiente de absorção da luz e do tamanho da amostra para estimar a integral são  $\beta_0 = 1$ ,  $\gamma = 1$  e  $N_{\text{amt}} \geq 50$  respectivamente.

# Capítulo 9

## Aplicações em Engenharia

Objetivando avaliar o desempenho do algoritmo MOFA em problemas de engenharia e estudar a influência do parâmetro de robustez sobre as soluções, neste capítulo são estudados problemas de projetos ótimos robustos realísticos com aplicações em projetos de estruturas, de construção de um robô industrial e de dinâmica de máquinas rotativas.

Em todos os problemas estudados não foram realizadas análises específicas sobre os parâmetros de incerteza de cada variável de projeto. Mais especificamente, não foram realizados experimentos para a verificação da amplitude do intervalo de variação dos parâmetros de incerteza, que regulam a inserção de robustez ao problema. Para a avaliação da influência do parâmetro de perturbação  $\delta$  sobre as curvas de Pareto, foram usados valores considerando a amplitude de variação das variáveis de projeto, de forma haver uma ponderação das amplitudes das variáveis. Por exemplo, considere um problema com três variáveis de projeto definidas nos intervalos:  $-1 \leq x_1 \leq 1$ ,  $-2 \leq x_2 \leq 2$  e  $0 \leq x_3 \leq 8$ . Observe que a amplitude de variação das variáveis  $x_1$ ,  $x_2$  e  $x_3$  são 2, 4 e 8, respectivamente. Neste caso, o vetor de perturbação será definido da seguinte forma:  $[\delta_1 \ \delta_2 \ \delta_3] = [\delta \ 2\delta \ 4\delta]$ , pois a amplitude do segundo intervalo é o dobro do primeiro intervalo e a amplitude do



terceiro intervalo é quatro vezes maior do que a do primeiro intervalo. Os valores do parâmetro  $\delta$  foram arbitrados como porcentagens da amplitude de variação da variável de projeto. Na situação hipotética considerada, os valores de  $\delta$  podem ser arbitrados como 5%, 10% ou 15% da amplitude de variação do intervalo da variável  $x_1$ , ou seja,  $\delta = 0,1$ ;  $\delta = 0,2$  ou  $\delta = 0,3$ .

A Tabela 9.1 apresenta os parâmetros do algoritmo MOFA usados nas simulações de todos os problemas estudados, exceto no problema de dinâmica de rotores, neste capítulo. O tamanho da população depende da quantidade de variáveis de cada problema.

Tabela 9.1: Parâmetros do algoritmo MOFA utilizados nas simulações.

<i>Parâmetros</i>	
$N_{ger}$	250
$\beta_0$	1
$\gamma$	1
$N_{amt}$	100
$r_p$	$10^3$

## 9.1 Problema de duas Barras

A estrutura geométrica do projeto de duas barras sujeito ao carregamento de uma força concentrada é apresentada na Fig. 9.1.

O problema de duas barras foi estudado originalmente por Palli et al. (1999). As variáveis de projeto são:  $y$  que representa a distância (em m) vertical entre B e C e  $x_1$  e  $x_2$  que representam as áreas da seção transversal das barras AC e BC em  $m^2$ , respectivamente. O problema de otimização, com dois objetivos, consiste na minimização da tensão nas barras AC e BC e da minimização do volume das mesmas.

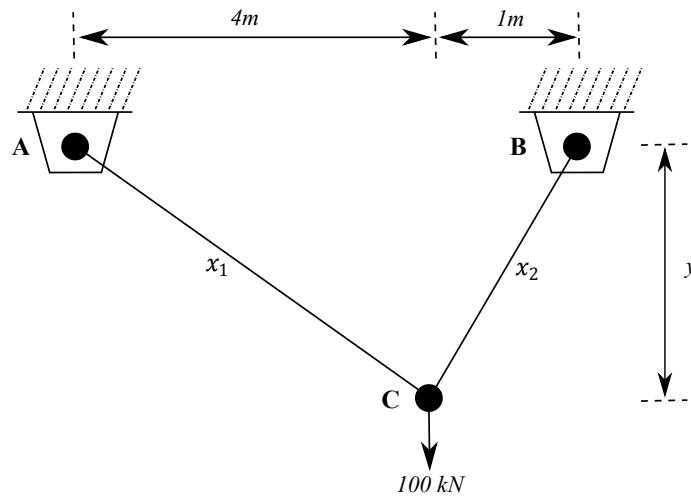


Figura 9.1: Geometria do problema das duas barras. Adaptado de (Deb,2001).

A formulação matemática para o problema de otimização é dada por:

$$\text{minimizar } f_1(x_1, x_2, y) = x_1\sqrt{16 + y^2} + x_2\sqrt{1 + y^2} \quad (9.1)$$

$$\text{minimizar } f_2(x_1, x_2, y) = \max(\sigma_{AC}, \sigma_{BC}) \quad (9.2)$$

$$\text{sujeito a } \max(\sigma_{AC}, \sigma_{BC}) \leq 10^5 \text{ Pa} \quad (9.3)$$

$$\left\{ \begin{array}{l} \sigma_{AC} = \frac{10\sqrt{16 + y^2}}{yx_1} \\ \sigma_{BC} = \frac{80\sqrt{1 + y^2}}{yx_2} \\ 0 \text{ m}^2 \leq x_1, x_2 \leq 0,01 \text{ m}^2 \text{ e } 1 \text{ m} \leq y \leq 3 \text{ m}. \end{array} \right.$$

Para avaliação da robustez foi utilizado o vetor de desvios  $[\delta_1 \ \delta_2 \ \delta_3] = [\delta \ \delta \ 20\delta]$ , para as variáveis  $x_1, x_2$  e  $y$  respectivamente, e os seguintes valores para o fator de perturbação  $\delta = 0,00075$ ;  $\delta = 0,001$  e  $\delta = 0,0015$ . O tamanho da população usada no processo de otimização é  $N_{\text{vag}} = 35$  vagalumes.

A Figura 9.2 abaixo, apresenta as curvas de Pareto nominal e robustas obtidas pelo

algoritmo MOFA para o problema de otimização de duas barras. Observa-se que à medida em que se aumenta o valor do parâmetro de perturbação  $\delta$ , o objetivo  $f_2$ , ou seja, o maior valor entre a tensão das barras  $AC$  e  $BC$ , diminui, apesar da variação da soma dos volumes das barras  $f_1$ , sofrer apenas um pequeno acréscimo. Isto mostra uma maior sensibilidade do objetivo  $f_2$  em relação à perturbação das variáveis de projeto para pequenos valores do volume das barras. Observa-se também que para valores do objetivo  $f_1$  maiores ou iguais a  $0,04\text{m}^3$  não há uma diminuição significativa dos valores da tensão nas barras, mostrando uma insensibilidade do objetivo  $f_2$  para valores do objetivo  $f_1$  na faixa citada.

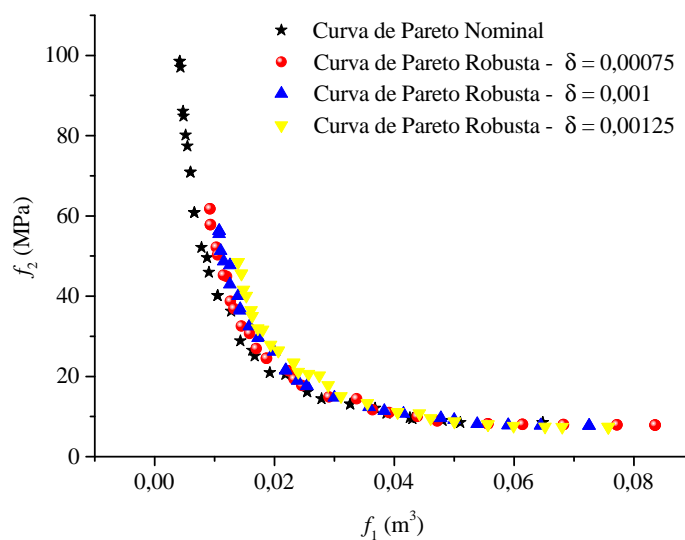


Figura 9.2: Curvas de Pareto robustas e nominal para o problema de duas barras.

As Tabelas 9.2, 9.3 e 9.4 apresentam valores das variáveis de projeto e dos objetivos para quatro pontos pertencentes a solução do problema, isto é, a curva de Pareto.

Tabela 9.2: Pontos da curva de Pareto nominal para o projeto ótimo de duas barras.

Parâmetro	Unidade	SOLUÇÃO			
$x_1$	m <sup>2</sup>	0,0006	0,0079	0,0044	0,0067
$x_2$	m <sup>2</sup>	0,0011	0,0079	0,0099	0,0099
$y$	m	1,690	1,590	2,787	2,974
$f_1$	m <sup>3</sup>	0,005	0,033	0,051	0,065
$f_2$	MPa	84,874	13,066	8,506	8,503

Tabela 9.3: Pontos da curva de Pareto robustas para o projeto ótimo de duas barras para  $\delta = 0,00075$ .

Parâmetro	Unidade	SOLUÇÃO			
$x_1$	m <sup>2</sup>	0,0005	0,0008	0,0023	0,0044
$x_2$	m <sup>2</sup>	0,0009	0,0023	0,0036	0,0098
$y$	m	1,981	1,845	1,814	2,033
$f_1$	m <sup>3</sup>	0,009	0,013	0,022	0,047
$f_2$	MPa	61,763	36,857	21,395	8,848

Tabela 9.4: Pontos da curva de Pareto robustas para o projeto ótimo de duas barras para  $\delta = 0,00125$ .

Parâmetro	Unidade	SOLUÇÃO			
$x_1$	m <sup>2</sup>	0,0005	0,0015	0,0027	0,0045
$x_2$	m <sup>2</sup>	0,0009	0,0025	0,0059	0,0099
$y$	m	1,799	1,904	2,067	2,949
$f_1$	m <sup>3</sup>	0,012	0,020	0,034	0,064
$f_2$	MPa	53,460	25,942	12,990	7,544

As Figuras 9.3, 9.4 e 9.5 abaixo ilustram a distribuição das variáveis de projeto para o problema de otimização robusta de duas barras para os seguintes valores do parâmetro de perturbação:  $\delta = 0,00075$  e  $\delta = 0,00125$ .

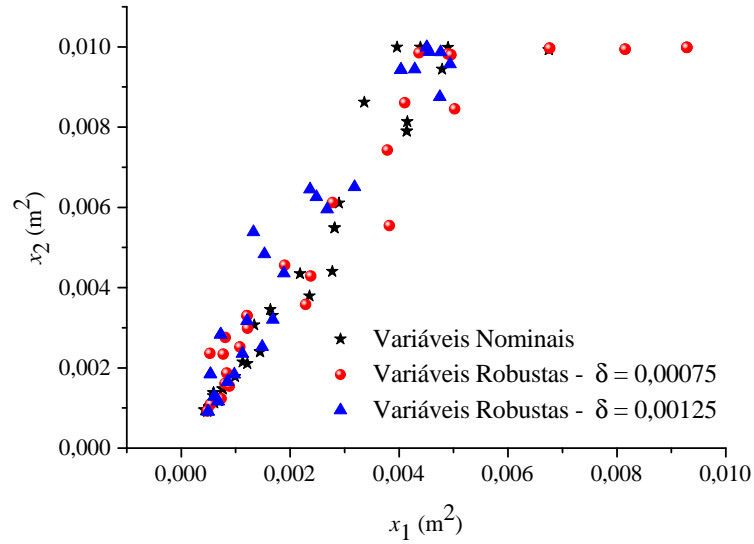


Figura 9.3: Valores das variáveis para os casos nominal e robusto para os parâmetros de robustez  $\delta = 0,00075$  e  $\delta = 0,00125$ .

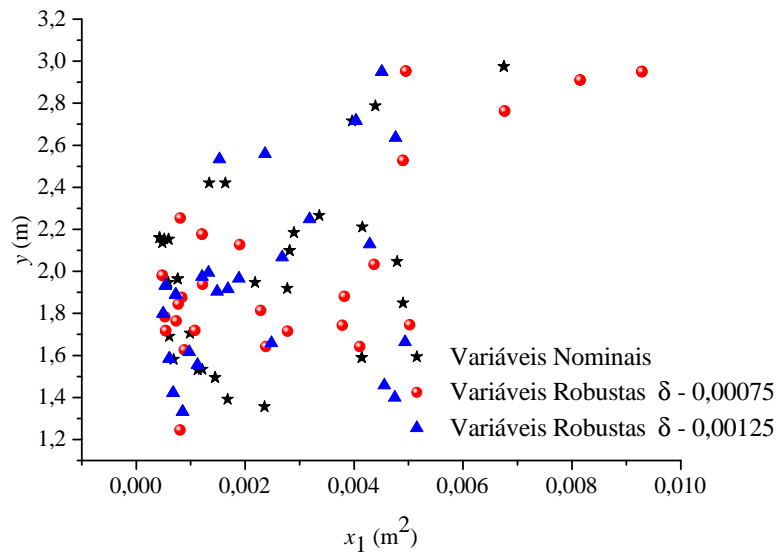


Figura 9.4: Valores das variáveis para os casos nominal e robusto para os parâmetros de robustez  $\delta = 0,00075$  e  $\delta = 0,00125$ .

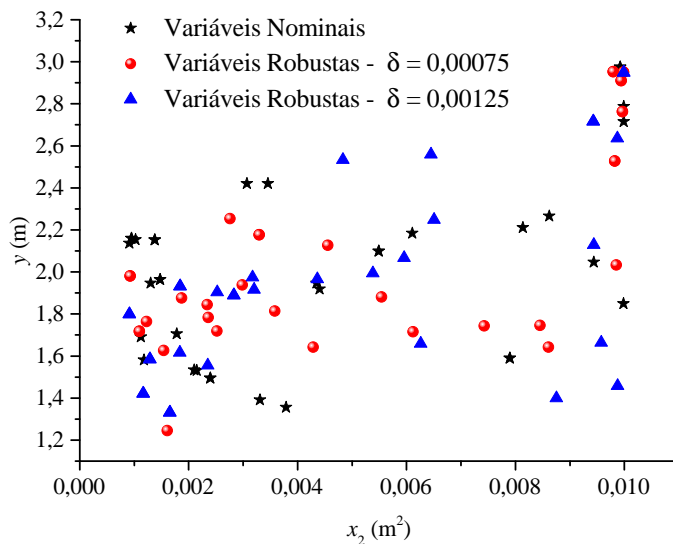


Figura 9.5: Valores das variáveis para os casos nominal e robusto para os parâmetros de robustez  $\delta = 0,00075$  e  $\delta = 0,00125$ .

## 9.2 Problema de uma Viga em Balanço

As vigas são elementos muito comuns em engenharia estrutural e podem possuir vários formatos. Este estudo de caso refere-se ao projeto ótimo robusto de uma viga em balanço. A estrutura geométrica do projeto de uma viga em balanço sujeita a aplicação de uma força estática concentrada em sua extremidade livre é apresentada na Fig. 9.6.

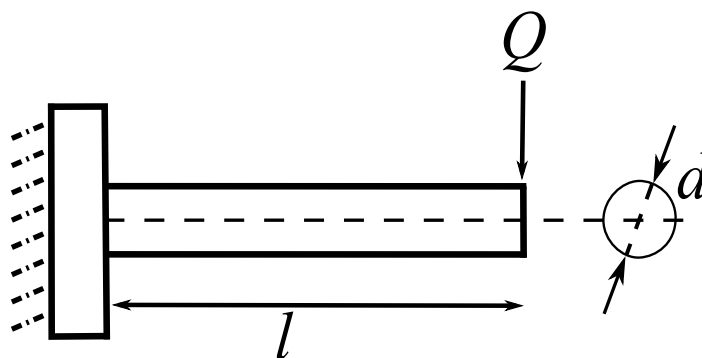


Figura 9.6: Geometria do problema da viga em balanço. Adaptada de (DEB, 2001).

As variáveis de projeto consideradas são o diâmetro  $d$  e o comprimento  $l$  da barra. O problema de otimização, com dois objetivos, consiste na minimização da deflexão da extremidade livre da viga e da minimização do peso da viga.

A formulação matemática para o problema de otimização é dada por:

$$\text{minimizar } f_1(d, l) = \rho \frac{\pi d^2}{4} l \quad (9.4)$$

$$\text{minimizar } f_2(d, l) = \frac{64Ql^3}{3E\pi d^4} \quad (9.5)$$

$$\text{sujeito a } \frac{32Ql}{\pi d^3} \leq S_y \quad (9.6)$$

$$\delta \leq \delta_{max} \quad (9.7)$$

onde,  $\delta_{max} = 5\text{mm}$ ,  $S_y = 300\text{MPa}$ ,  $Q = 1\text{kN}$ ,  $\rho = 7800\text{kg/m}^3$ ,  $E = 207\text{GPa}$  e  $10\text{ mm} \leq d \leq 50\text{mm}$ ,  $200\text{ mm} \leq l \leq 1000\text{ mm}$ . Convém observar que as equações apresentadas na formulação do problema de otimização são válidas somente para este formato de viga, ou seja, de seção circular.

A primeira restrição refere-se à tensão máxima admissível e a segunda restrição refere-se à deflexão máxima permitida no projeto. Observa-se teoricamente que, se houver a diminuição das dimensões da viga, o peso da mesma será reduzido, porém o decréscimo das dimensões acarreta no aumento da deflexão final. Assim, este problema possui objetivos conflitantes.

Para avaliação da robustez foi utilizado o vetor de desvios  $[\delta_1 \ \delta_2] = [\delta \ 20\delta]$ , para as variáveis  $d$  e  $l$  respectivamente, e os seguintes valores para o fator de perturbação  $\delta = 0,0009$ ,  $\delta = 0,0011$  e  $\delta = 0,0013$ . O tamanho da população é  $N_{vag} = 25$  vagalumes.

A Figura 9.7, apresenta a curva de Pareto nominal e mostra também o efeito do parâmetro  $\delta$  sobre as curvas de Pareto robustas obtidas pelo algoritmo MOFA. Observa-se que, para vigas com peso inferior a 2 kg, o aumento do parâmetro de robustez  $\delta$

promove o distanciamento entre as curvas robustas e este comportamento caracteriza uma maior sensibilidade do objetivo  $f_2$  (deflexão) para vigas com peso inferior a 2 kg quando comparado com vigas com peso entre 2 kg e 4 kg. Observa-se também que não há uma diferença significativa entre os valores da deflexão final para vigas com peso superior a 2 kg, ou seja o objetivo  $f_2$  é pouco sensível nesta região.

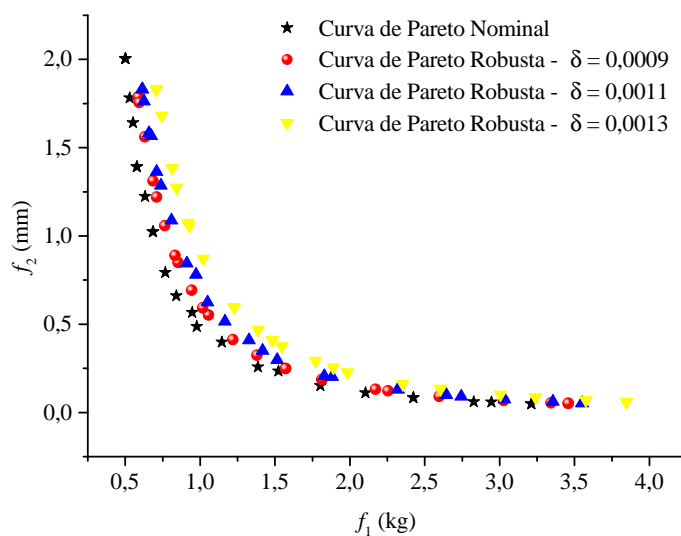


Figura 9.7: Curvas de Pareto robustas e nominal para o problema da viga em balanço obtidas pelo MOFA.

As Tabelas 9.5, 9.6 e 9.7 apresentam valores das variáveis de projeto e dos objetivos para quatro pontos pertencentes à solução do problema.

Tabela 9.5: Pontos da curva de Pareto nominal para o projeto ótimo de uma viga em balanço.

Parâmetro	Unidade	SOLUÇÃO			
$d$	mm	49,797	33,364	24,354	19,349
$l$	mm	200	200	200	200
$f_1$	kg	3,170	1,431	0,770	0,488
$f_2$	mm	0,046	0,224	0,787	1,943



Tabela 9.6: Pontos da curva de Pareto robusta com  $\delta = 0,0009$  para o projeto ótimo de uma viga em balanço.

Parâmetro	Unidade	SOLUÇÃO			
$d$	mm	49,884	34,383	24,666	19,924
$l$	mm	200	200	200	200
$f_1$	kg	3,451	1,668	0,875	0,579
$f_2$	mm	0,051	0,223	0,811	1,840

Tabela 9.7: Pontos da curva de Pareto robusta com  $\delta = 0,0013$  para o projeto ótimo de uma viga em balanço.

Parâmetro	Unidade	SOLUÇÃO			
$d$	mm	49,994	32,857	23,558	20,169
$l$	mm	200	200	200	201
$f_1$	kg	3,647	1,619	0,857	0,643
$f_2$	mm	0,056	0,285	1,013	1,863

A Figura 9.8 abaixo apresenta, no espaço de projetos, valores das variáveis para o caso nominal e robusto, para os seguintes valores do parâmetro de robustez  $\delta = 0,0009$  e  $\delta = 0,0013$ .

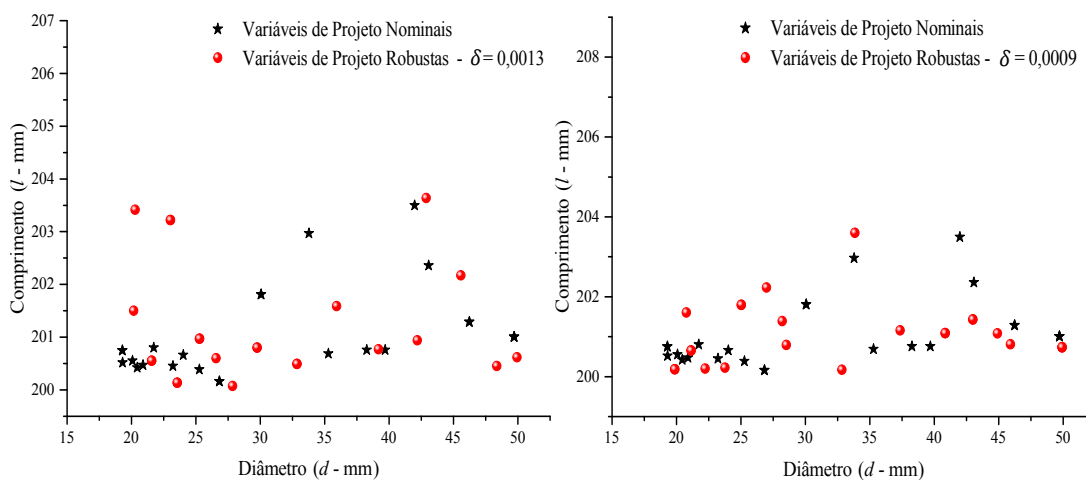


Figura 9.8: Valores das variáveis para os casos nominal e robusto para os parâmetros de robustez  $\delta = 0,0009$  e  $\delta = 0,0013$ .

### 9.3 Problema da Viga Soldada

O próximo estudo de caso consiste no projeto de uma viga engastada de secção retangular submetida a uma força  $F$  em sua extremidade livre. A viga precisa ser soldada em uma estrutura e o problema consiste na minimização do custo e da deflexão final. A estrutura geométrica do problema é apresentada na Fig. 9.9.

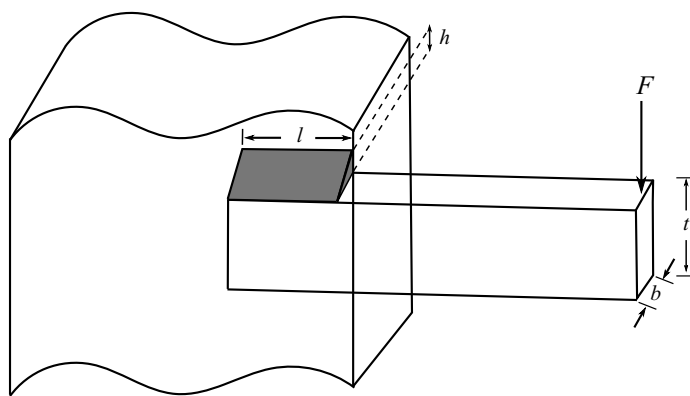


Figura 9.9: Geometria do problema da viga soldada. Adaptada de (Castro, 2001).

As quatro variáveis de projeto consideradas são a espessura e o comprimento do cordão de solda, representadas respectivamente por  $h$  e o comprimento  $l$ , largura e altura da secção retangular da viga, representadas respectivamente por  $b$  e  $t$ . A formulação matemática para o problema de otimização, válida apenas para a geometria considerada na

Fig. 9.9, é dada por:

$$\text{minimizar } f_1(h, l, t, b) = 1,1047h^2l + 0,04811tb(14 + l) \quad (9.8)$$

$$\text{minimizar } f_2(h, l, t, b) = \frac{2.1952}{t^3b} \quad (9.9)$$

$$\text{sujeito a } \tau - 13600 \leq 0$$

$$\sigma - 30000 \leq 0$$

$$6000 - P_c \leq 0$$

$$h - b \leq 0$$

$$\frac{2.1952}{t^3b} - 0,010 \leq 0$$

onde

$$\left\{ \begin{array}{l} \tau = \sqrt{(\tau')^2 + (\tau'')^2 + (l\tau'\tau'')/\sqrt{0,25(l^2 + (h+t)^2)}}; \\ \tau' = \frac{6000}{\sqrt{2hl}}, \quad \tau'' = \frac{6000(14 + 0,5l)\sqrt{l^2 + (h+t)^2}}{2[0,707hl(l^2/12 + 0,25(h+t)^2)]}; \\ \sigma = \frac{504000}{t^2b}; \\ P_c = 64746,022(1 - 0,0282346)tb^3 \\ 0,125 \text{ m} \leq h, b \leq 5 \text{ m} \quad \text{e} \quad 0,1 \text{ m} \leq l, t \leq 10 \text{ m}. \end{array} \right.$$

Os valores 1,1047 e 0,04811 na Eq. 9.8 estão relacionados ao custo do material por unidade de volume. Os outros coeficientes que aparecem nas equações são combinações dos valores do módulo de elasticidade do material, da carga aplicada e do comprimento da viga. As duas primeiras restrições garantem que a tensão de cisalhamento e a tensão normal desenvolvidas ao longo do apoio da viga sejam menores que suas respectivas tensões admissíveis, inerentes ao material. A terceira restrição garante que o esforço resistente (ao longo da direção da altura da viga) seja maior que a carga  $F$  aplicada na extremidade. A quarta restrição serve para garantir que a deflexão final não seja maior que um valor pre-

estabelecido, ou seja, é um limite máximo de deslocamento da viga. A última restrição é apenas para garantir que a espessura da solda não seja maior que a largura da viga.

Para avaliação da robustez foi utilizado o vetor de desvios  $[\delta_1 \delta_2 \delta_3 \delta_4] = [\delta \ 2\delta \ 2\delta \ \delta]$ , para as variáveis  $h, l, t$  e  $b$  respectivamente, e os seguintes valores para o fator de perturbação  $\delta = 0,05$ ,  $\delta = 0,1$  e  $\delta = 0,15$ . O tamanho da população considerada é de  $N_{\text{vag}} = 45$  vagalumes e o fator de penalidade  $r_p$  igual a  $10^3$ .

A Figura 9.10 abaixo, ilustra o comportamento de duas curvas robustas ( $\delta = 0,05$ ,  $\delta = 0,10$  e  $\delta = 0,15$ ) em relação à curva nominal (sem robustez) obtidas pelo algoritmo MOFA.

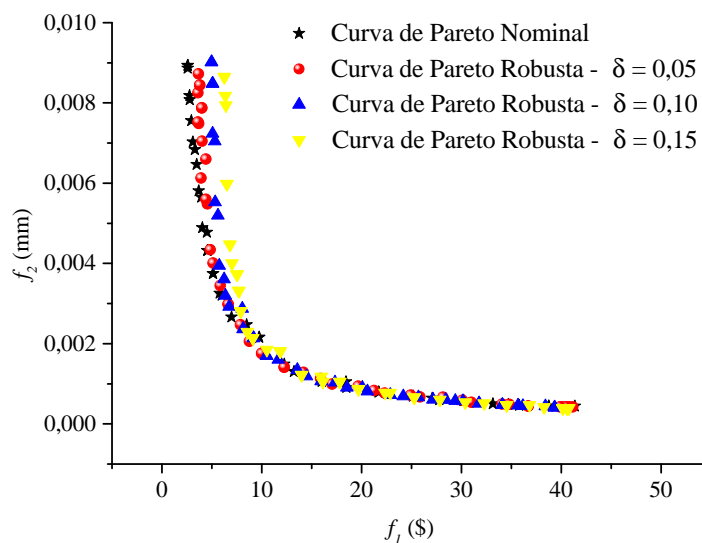


Figura 9.10: Curvas de Pareto nominal e robustas para o problema da viga soldada.

Observe que para valores maiores das variáveis de projeto (maior custo) há pouca ou quase nenhuma sensibilidade dos objetivos, porém para pequenos valores das variáveis (menor custo) há uma maior sensibilidade do objetivo  $f_2$  (Deflexão). Esta sensibilidade é caracterizada pelo distanciamento entre as curvas de Pareto nominal e robustas, para pequenos valores das variáveis de projeto.

As Tabelas 9.8, 9.9 e 9.10 apresentam alguns pontos da curva solução nominal e robustas, com  $\delta = 0, 10$  e  $\delta = 0, 15$ , obtidas pelo algoritmos MOFA.

Tabela 9.8: Pontos da curva de Pareto nominal para o projeto ótimo de uma viga soldada.

Parâmetro	Unidade	SOLUÇÃO			
$h$	m	0,258	0,329	0,845	1,646
$l$	m	4,849	4,058	1,288	0,582
$t$	m	9,555	9,994	9,944	9,997
$b$	m	0,262	0,718	2,102	2,446
$f_1$	\$	2,626	6,724	16,395	18,898
$f_2$	mm	9,613	3,060	1,061	0,898

Tabela 9.9: Pontos da curva de Pareto robusta com  $\delta = 0, 1$  para o projeto ótimo de uma viga soldada.

Parâmetro	Unidade	SOLUÇÃO			
$h$	m	0,295	0,405	0,679	0,737
$l$	m	4,399	2,810	1,974	1,357
$t$	m	8,698	9,783	9,864	9,880
$b$	m	0,445	0,590	2,049	2,399
$f_1$	\$	5,149	6,498	18,297	20,065
$f_2$	mm	5,783	3,221	1,003	0,859

Tabela 9.10: Pontos da curva de Pareto robusta com  $\delta = 0, 15$  para o projeto ótimo de uma viga soldada.

Parâmetro	Unidade	SOLUÇÃO			
$h$	m	0,332	0,382	1,019	1,163
$l$	m	4,679	3,143	0,844	0,736
$t$	m	7,646	8,402	9,993	9,950
$b$	m	0,569	0,697	1,854	2,085
$f_1$	\$	6,537	7,291	16,767	18,551
$f_2$	mm	6,185	3,981	1,008	0,914

As Figuras 9.11, 9.12, 9.13, 9.14, 9.15 e 9.16 abaixo, apresentam as relações existentes entre as variáveis de projeto para os casos nominal e robustos com  $\delta = 0, 10$  e  $\delta = 0, 15$ , ou seja, apresentam as seguintes relações  $h \times l, h \times t, h \times b, l \times t, l \times b$  e  $t \times b$ .

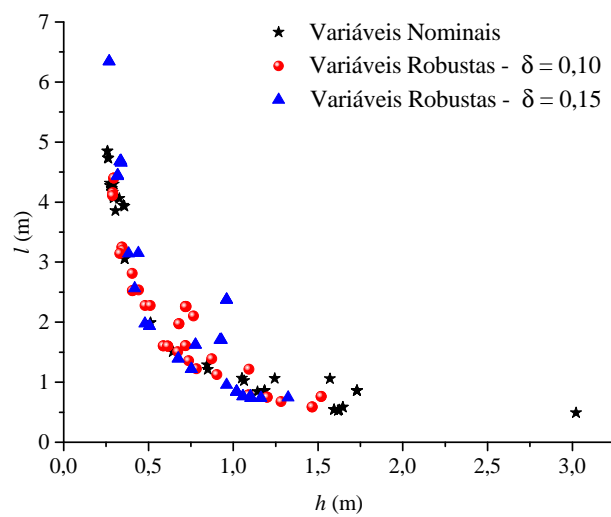


Figura 9.11: Variáveis  $h$  e  $l$  para os casos nominal e robusto ( $\delta = 0, 10, \delta = 0, 15$ ).

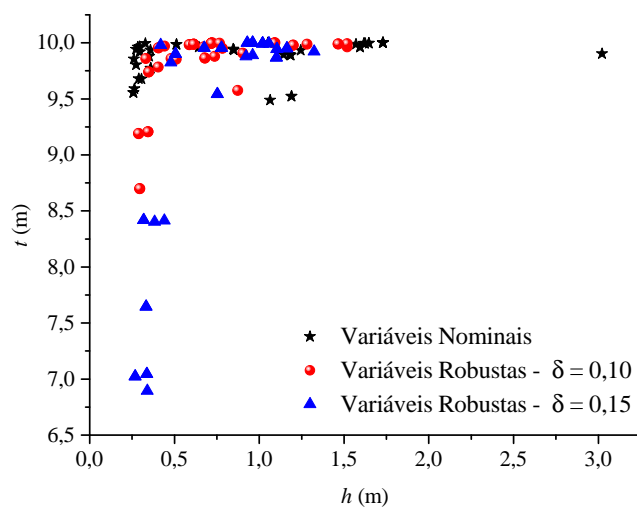


Figura 9.12: Variáveis  $h$  e  $t$  para os casos nominal e robusto ( $\delta = 0, 10, \delta = 0, 15$ ).

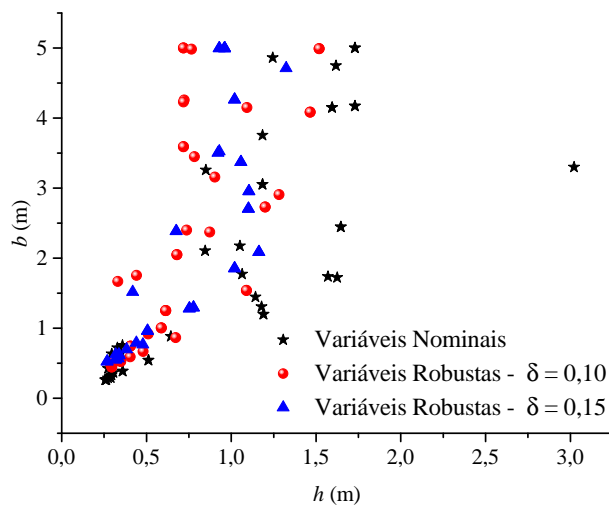


Figura 9.13: Variáveis  $h$  e  $b$  para os casos nominal e robusto ( $\delta = 0, 10, \delta = 0, 15$ ).

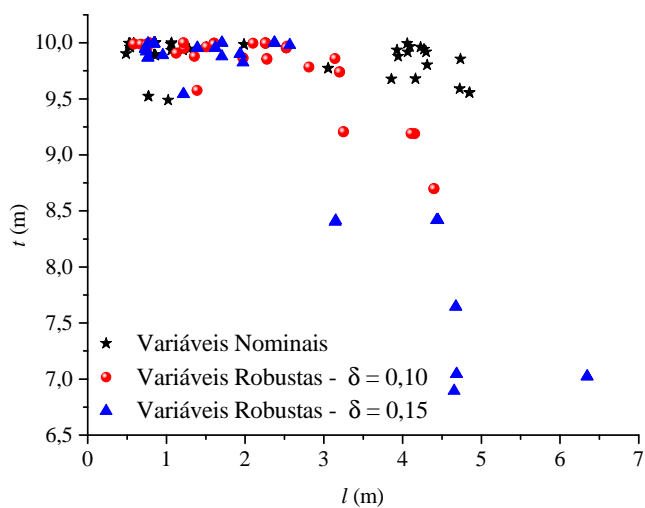


Figura 9.14: Variáveis  $l$  e  $t$  para os casos nominal e robusto ( $\delta = 0, 10, \delta = 0, 15$ ).

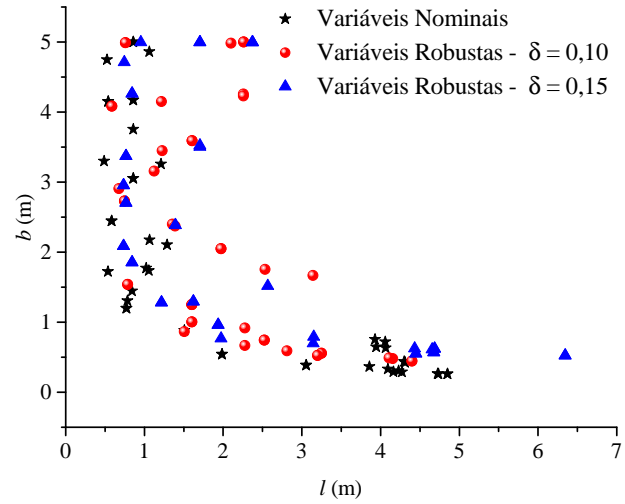


Figura 9.15: Variáveis  $l$  e  $b$  para os casos nominal e robusto ( $\delta = 0, 10, \delta = 0, 15$ ).

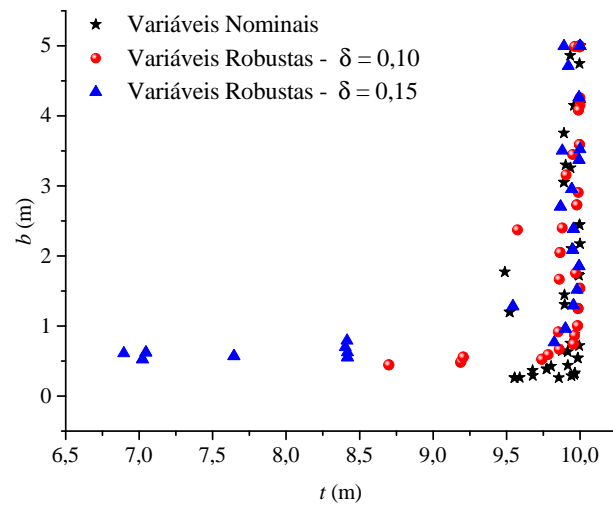


Figura 9.16: Variáveis  $t$  e  $b$  para os casos nominal e robusto ( $\delta = 0, 10, \delta = 0, 15$ ).



## 9.4 Projeto Ótimo Robusto de um Robô Industrial utilizando Metamodelos

### 9.4.1 Metamodelagem

As características dos problemas reais em engenharia, tais como geometrias complexas, não linearidades e presença do regimes de turbulência, contribuem para um alto custo computacional relacionado a simulação, devido ao refinamento e/ou atualização da malha utilizada, o que é mais evidente em projetos de otimização (MALISKA, 2004).

Segundo Santos (2002) a modelagem de processos reais é de grande importância, pois podem haver várias limitações proibitivas de experimentação do sistema real.

Apesar do modelo de simulação ser bem mais simples do que o sistema real, este pode ser bastante complicado em termos de custo computacional, pois o modelo de simulação requer a realização de inúmeras execuções do programa de simulação para um grande número de combinações das variáveis de entrada (LOBATO, 2008). Sendo assim, como alternativa a este inconveniente, pode-se construir uma relação matemática simplificada do modelo de simulação que relaciona as variáveis de entrada com as saídas do modelo através de um número relativamente baixo de experimentos (MYERS, 1971; BOX et al., 1978). Esta relação matemática que simplifica o modelo de simulação é denominada de metamodelo (BARTON, 1992). Segundo Frisso, Scarpel e Ferrari (2011) diversas aplicações de otimização baseada em metamodelos podem ser encontradas na indústria aeronáutica, aeroespacial, automotiva, entre outras.

Devido a sua simplicidade conceitual, os metamodelos mais empregados na literatura são baseados em aproximações polinomiais e são denominados de superfícies de respostas, sendo que as aproximações mais empregadas são aquelas que possuem termos lineares,

quadráticos e cúbicos (MYERS, 1971; BOX et al., 1978):

$$Y = \beta_0 + \sum_{i=1}^n \beta_i x_i + \sum_{i=1}^{n-1} \sum_{j \neq i, j > i}^n \beta_{i+j+1} x_i x_j + \sum_{i=1}^n \beta_{2n+i} x_i^2 + \sum_{i=1}^n \beta_{3n+i} x_i^3 \quad (9.10)$$

Para construir metamodelos existem alternativas paramétricas, tais como regressão polinomial e Kriging, e não paramétricas tais como regressão *projection-pursuit* e funções de base radial. Na forma paramétrica, assume-se que a forma funcional da relação entre a resposta e as variáveis é conhecida, enquanto a forma não paramétrica usa diferentes tipos de modelos locais simples para construir o modelo geral. Fang et al. (2005) e Viana (2008) realizam um estudo abrangente sobre técnicas de metamodelagem e aplicações.

Um ponto importante quando se trata de meta-modelos tem a ver com os critérios que garantem a confiabilidade do metamodelo utilizado. A garantia de uma boa aproximação depende de uma análise de parâmetros estatísticos obtidos através dos dados experimentais e do metamodelo considerado. Neste contexto, são descritas os principais operadores estatísticos utilizados para descrever o quanto o meta-modelo empregado é confiável em comparação com a situação real (MYERS, 1971; BOX et al., 1978; VIANA, 2008; LOBATO, 2008):

- $F_{stat} - F$  estatístico:

$$F_{stat} = \frac{(SST - SSE)/p}{SSE/(n - p - 1)} \quad (9.11)$$

onde  $SSE = \sum_{i=1}^n (f_i - f'_i)^2$ ,  $SST = \sum_{i=1}^n (f_i - \bar{f})^2$ ,  $n$  é o número de experimentos,  $p$  é o número de termos não constantes do metamodelo,  $f_i$  é o valor da resposta dos experimentos,  $f'_i$  é o valor de resposta medido a partir do metamodelo e  $\bar{f}$  é a média das respostas  $f$ .

- $R^2$  - coeficiente de correlação dos dados:

$$R^2 = 1 - \frac{SSE}{SST} \quad (9.12)$$

- $R_{adj}^2$  - coeficiente de correlação ajustado:

$$R_{adj}^2 = 1 - (1 - R^2) \frac{n - 1}{n - p - 1} \quad (9.13)$$

- $RMSE$  - erro quadrático médio:

$$RMSE = \sqrt{\frac{SSE}{n - p - 1}} \quad (9.14)$$

No contexto da metamodelagem, a metodologia empregada para resolução de problemas de otimização é apresentada na Fig. 9.17 a seguir:

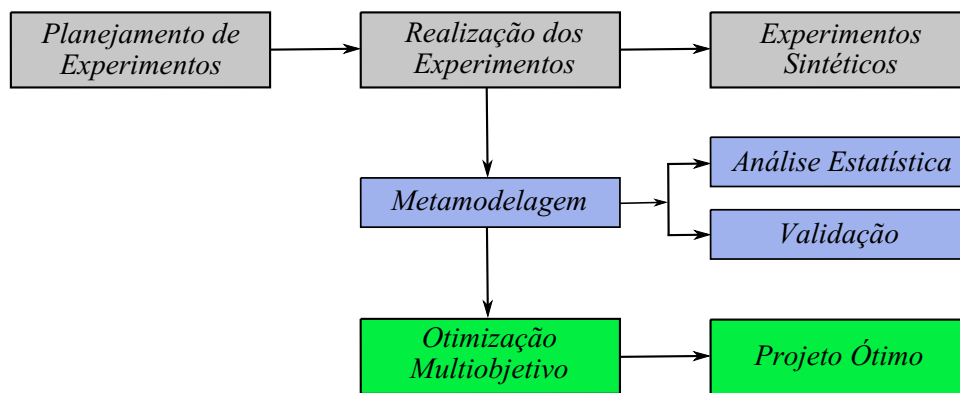


Figura 9.17: Etapas da resolução de problemas reais utilizando metamodelos. Adaptado de (LOBATO, 2008).

Segundo Viana (2008), as etapas da metodologia de resolução de problemas de otimização multiobjetivo utilizando metamodelos é descrita por:

- i. escolha das variáveis independentes e a elaboração do planejamento dos experimentos;
- ii. escolha do metamodelo a ser empregado para cada uma das saídas observadas (variáveis dependentes);
- iii. realização dos experimentos (em laboratório ou a partir de um modelo de simulação);
- iv. cálculo dos coeficientes de cada metamodelo empregado;
- v. análise estatística para validação dos resultados obtidos;
- vi. formulação do problema de otimização multiobjetivo.

Objetivando a aplicação da metodologia descrita nesta seção, será estudado um problema de otimização, com três objetivos, de um robô industrial com um mecanismo de balanceamento usando uma mola.

#### **9.4.2 Projeto Ótimo Robusto de um Robô Industrial**

A Figura abaixo ilustra um modelo de um robô no qual um mecanismo de mola hidráulica é utilizado para balanceamento do braço AB (ESCHENAUER et al., 1990).

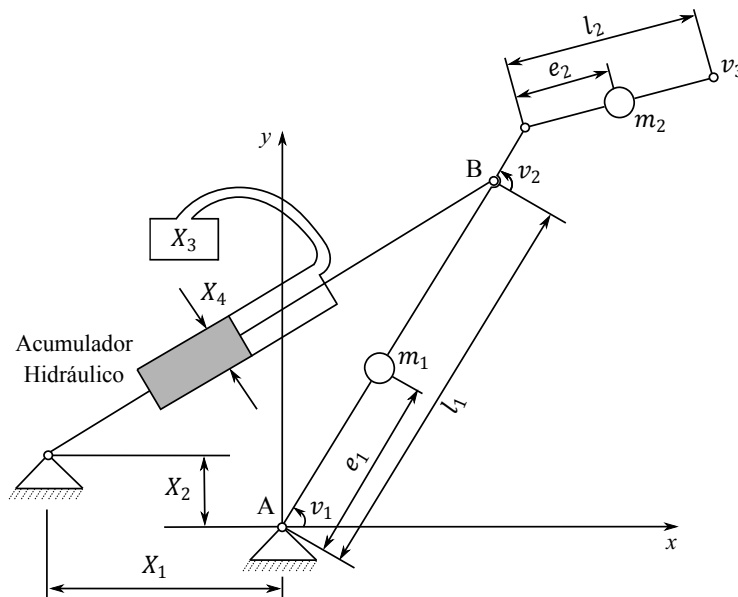


Figura 9.18: Robô industrial com um mecanismo de mola hidráulica. Reproduzida de (ESCHENAUER, 1990).

As variáveis de projeto que influenciam o balanceamento do braço do robô são: localização do mecanismo de mola em relação à base do robô ( $X_1$  e  $X_2$  em metros), a pressão do acumulador hidráulico ( $X_3$  em MPa) e o diâmetro do cilindro ( $X_4$  em metros). O desempenho do robô é função dos seguintes critérios: torque estático na haste do motor de direção, o trabalho executado pelo motor e a força estática na junta A, representados respectivamente por  $y_1$ ,  $y_2$  e  $y_3$ .

Será utilizado um metamodelo com termos quadráticos para a obtenção dos critérios  $y_i^{met}$  ( $i = 1, 2, 3$ ) dado, de forma genérica, por:

$$Y = \beta_0 + \beta_1 x_1 + \cdots + \beta_4 x_4 + \beta_5 x_1^2 + \cdots + \beta_8 x_4^2 + \beta_9 x_1 x_2 + \beta_{10} x_1 x_3 + \quad (9.15)$$

$$+ \beta_{11} x_1 x_4 + \beta_{12} x_2 x_3 + \beta_{13} x_2 x_4 + \beta_{14} x_3 x_4$$

onde os  $\beta_i$  são os coeficientes da aproximação polinomial ( $i = 1, \dots, 14$ ),  $x_i, i = 1, \dots, 4$

são as variáveis de projetos (independentes) codificadas e  $Y = [y_1^{met} \ y_2^{met} \ y_3^{met}]$  é o vetor de respostas do metamodelo.

#### Planejamento de Experimentos

Neste estudo de caso não foram realizadas simulações, e sim usados os dados apresentados na tese de Lobato (2008) que por sua vez os coletou do trabalho de Eschenauer et al. (1990). Para que as variáveis estejam na mesma ordem de grandeza, foram consideradas as seguintes codificações:

$$x_1 = \frac{X_1}{0,01}, \quad x_2 = \frac{X_2 - 0,30}{0,04}, \quad x_3 = \frac{X_3 - 3,3}{0,3}, \quad x_4 = \frac{X_4 - 0,037}{0,003} \quad (9.16)$$

Os parâmetros do robô industrial usados nas simulações estão presentes na Tab. 9.11 e os valores do planejamento de experimentos (variáveis de projeto) e as respectivas saídas encontram-se na Tab. 9.12.

Tabela 9.11: Parâmetros do robô industrial usados nas simulações (ESCHENAUER et al., 1990).

	Descrição dos símbolos	Valor	Unidade
$m_1$	massa do braço	254,4	kg
$m_2$	massa do antebraço	105,3	kg
$l_1$	comprimento do braço	1,0	m
$l_2$	comprimento do antebraço	1,35	m
$e_1$	distância do centro de gravidade até o braço	0,52	m
$e_2$	distância do centro de gravidade até o antebraço	0,53	m
$v_3$	carregamento máximo de carga	80	kg
	número de molas	2	-

Para o ajuste de cada metamodelo, Lobato (2008) utilizou o método de otimização Programação Quadrática Sequencial com 10000 configurações iniciais de projeto, geradas de forma aleatória no seguinte espaço de projetos  $-10000 \leq \beta_i \leq 10000$  ( $i = 1, \dots, 14$ ). Os coeficientes e os valores das estatísticas, obtidas pelo autor citado acima, usadas para

Tabela 9.12: Planejamento de experimentos e respostas obtidas nas simulações para o projeto ótimo robusto de um robô industrial (ESCHENAUER et al., 1990).

Experimento	$x_1$	$x_2$	$x_3$	$x_4$	$X_1$	$X_2$	$X_3$	$X_4$	$y_1$	$y_2$	$y_3$
1	1	1	1	1	0,01	0,34	3,6	0,040	3546,7	16652,3	4405,4
2	-1	1	1	1	-0,01	0,34	3,6	0,040	3684,1	17443,4	4552,4
3	1	-1	1	1	0,01	0,26	3,6	0,040	1960,5	13648,9	3042,8
4	-1	-1	1	1	-0,01	0,26	3,6	0,040	1727,8	14200,7	2736,7
5	1	1	-1	1	0,01	0,34	3,0	0,040	2606,3	13788,4	3510,8
6	-1	1	-1	1	-0,01	0,34	3,0	0,040	2720,9	14447,8	3611,5
7	1	-1	-1	1	0,01	0,26	3,0	0,040	1828,5	11856,7	3433,9
8	-1	-1	-1	1	-0,01	0,26	3,0	0,040	1746,0	11884,0	3168,5
9	1	1	1	-1	0,01	0,34	3,6	0,034	2053,8	10851,3	3034,0
10	-1	1	1	-1	-0,01	0,34	3,6	0,034	1886,5	10851,3	2824,2
11	1	-1	1	-1	0,01	0,26	3,6	0,034	2452,1	10581,8	3895,2
12	-1	-1	1	-1	-0,01	0,26	3,6	0,034	2438,7	10851,8	3752,3
13	1	1	-1	-1	0,01	0,34	3,0	0,034	1977,5	9761,6	3456,0
14	-1	1	-1	-1	-0,01	0,34	3,0	0,034	1954,6	9761,6	3205,6
15	1	-1	-1	-1	0,01	0,26	3,0	0,034	2827,7	9762,0	4270,2
16	-1	-1	-1	-1	-0,01	0,26	3,0	0,034	2816,5	9762,0	4169,2
17	1	0	0	0	0,01	0,30	3,3	0,037	1880,4	11411,5	3178,2
18	-1	0	0	0	-0,01	0,30	3,3	0,037	1694,0	11411,5	2857,2
19	0	1	0	0	0,00	0,34	3,3	0,037	2154,7	11658,6	3053,5
20	0	-1	0	0	0,00	0,26	3,3	0,037	2084,2	11409,8	3528,7
21	0	0	1	0	0,00	0,30	3,6	0,037	1935,6	12054,9	2813,7
22	0	0	-1	0	0,00	0,30	3,0	0,037	1785,7	10764,7	3248,1
23	0	0	0	1	0,00	0,30	3,3	0,040	2087,1	13966,0	3066,5
24	0	0	0	-1	0,00	0,30	3,3	0,034	2189,5	10305,7	3526,9

validar o metamodelo são apresentados na Tab. 9.13.

Observa-se, pelos parâmetros estatísticos da Tab. 9.13, que os metamodelos representam satisfatoriamente os critérios a serem otimizados no espaço de projetos considerado.

#### Projeto ótimo robusto de um robô industrial

A formulação matemática para o problema de otimização multiobjetivo é dada por (ES-

CHENAUER et al., 1990):

$$\text{minimizar } f_1(x_1, x_2, x_3, x_4) = y_1^{met} \quad (9.17)$$

$$\text{minimizar } f_2(x_1, x_2, x_3, x_4) = y_2^{met} \quad (9.18)$$

$$\text{minimizar } f_3(x_1, x_2, x_3, x_4) = y_3^{met} \quad (9.19)$$

onde  $-1 \leq x_i \leq 1$ ,  $i = 1, \dots, 4$ .

Tabela 9.13: Coeficientes da aproximação polinomial e parâmetros estatísticos para o projeto ótimo de um robô industrial (LOBATO, 2008).

Coeficiente	$y_1^{met}$	$y_2^{met}$	$y_3^{met}$
$\beta_0$	1838,779	11416,08	3023,202
$\beta_1$	25,80000	-127,7555	74,93887
$\beta_2$	150,1722	626,9777	-19,11667
$\beta_3$	79,00558	852,6444	-56,50556
$\beta_4$	72,83334	1966,616	-33,61668
$\beta_5$	-51,57915	-4,583422	-5,502169
$\beta_6$	280,6708	131,6167	267,8979
$\beta_7$	21,87080	-6,283291	7,697942
$\beta_8$	299,5208	719,7667	273,4979
$\beta_9$	-25,10001	-37,58753	-37,68124
$\beta_{10}$	17,12497	-57,88750	-0,268739
$\beta_{11}$	-9,474989	-109,9749	-23,76873
$\beta_{12}$	159,4625	126,2749	165,4312
$\beta_{13}$	497,3625	654,5875	454,0812
$\beta_{14}$	172,6625	367,4625	162,9937
$R^2$	0,973316	0,990621	0,873555
$R_{adj}^2$	0,931808	0,976032	0,696863
$F_{stat}$	23,448848	67,902679	4,441239
$RMSE$	141,62452	328,28349	295,00784

Para avaliação da robustez foi utilizado o vetor de desvios  $[\delta_1 \delta_2 \delta_3 \delta_4] = [\delta \delta \delta \delta]$  e os seguintes valores para o fator de perturbação  $\delta = 0, 1$  e  $\delta = 0, 2$ . O tamanho da população considerada é de  $N_{vag} = 45$  vagalumes. As Figuras 9.19, 9.20 e 9.21 apresentam as soluções nominal e robustas, obtidas pelo algoritmo MOFA.



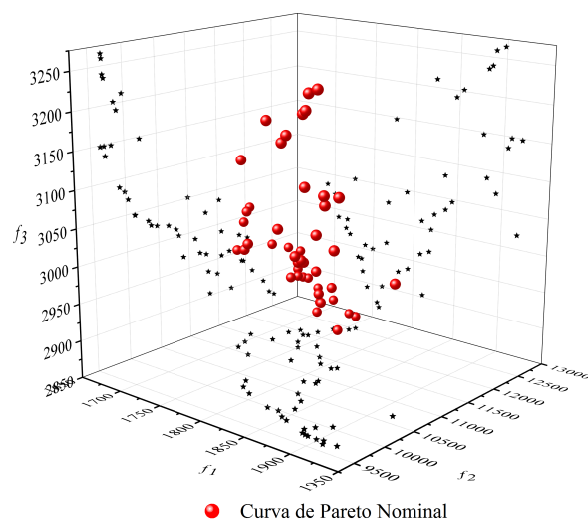


Figura 9.19: Curva de Pareto nominal para o projeto ótimo de um robô industrial.

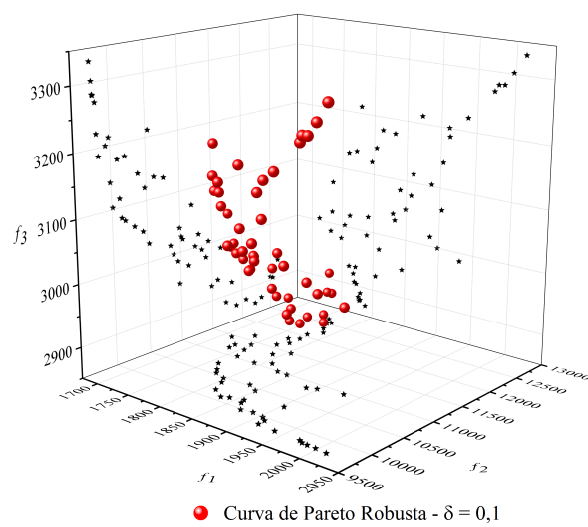


Figura 9.20: Curva de Pareto robusta ( $\delta = 0,1$ ) para o projeto ótimo de um robô industrial.

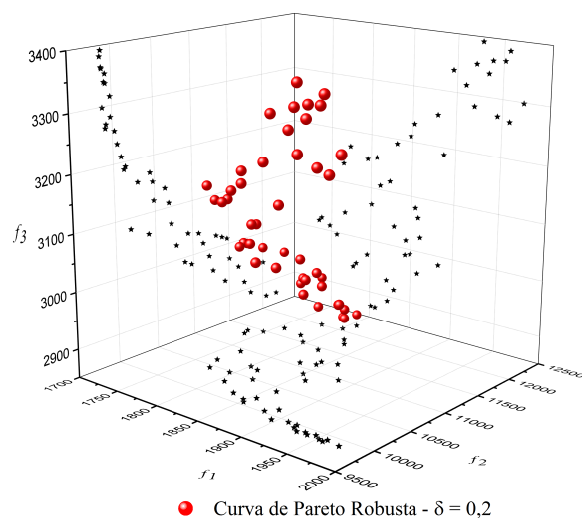


Figura 9.21: Curva de Pareto robusta ( $\delta = 0,2$ ) para o projeto ótimo de um robô industrial.

As Figuras que se seguem, ilustram as relações  $f_1 \times f_2$ ,  $f_1 \times f_3$  e  $f_2 \times f_3$  para o valor do parâmetro de perturbação  $\delta = 0,1$ .

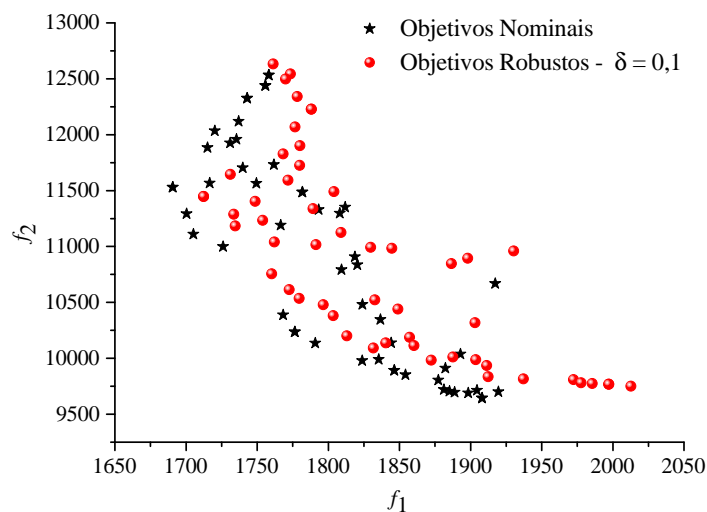


Figura 9.22:  $y_1^{met} \times y_2^{met}$  para o problema do projeto ótimo robusto ( $\delta = 0,1$ ) de um robô industrial.

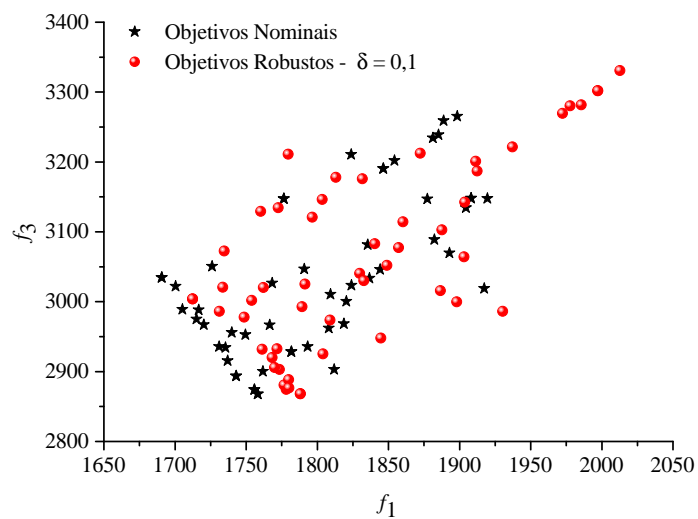


Figura 9.23:  $y_1^{met} \times y_3^{met}$  para o problema do projeto ótimo robusto ( $\delta = 0,1$ ) de um robô industrial.

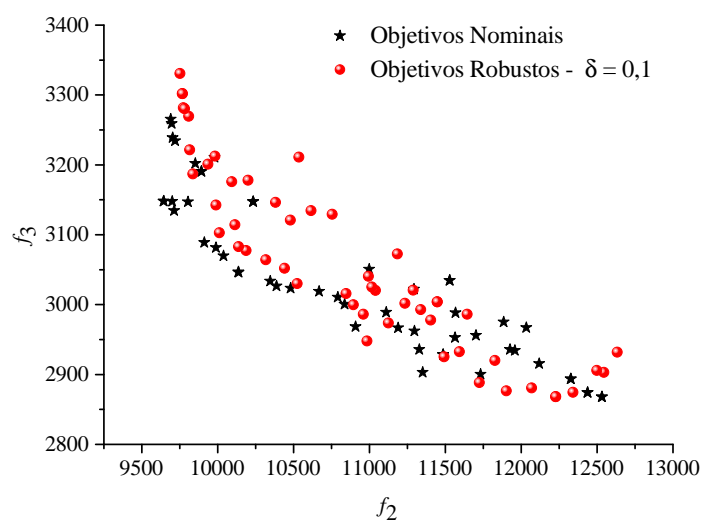


Figura 9.24:  $y_2^{met} \times y_3^{met}$  para o problema do projeto ótimo robusto ( $\delta = 0,1$ ) de um robô industrial.

As Figuras a seguir ilustram as relações  $f_1 \times f_2$ ,  $f_1 \times f_3$  e  $f_2 \times f_3$  para o valor do parâmetro de perturbação  $\delta = 0,2$ .

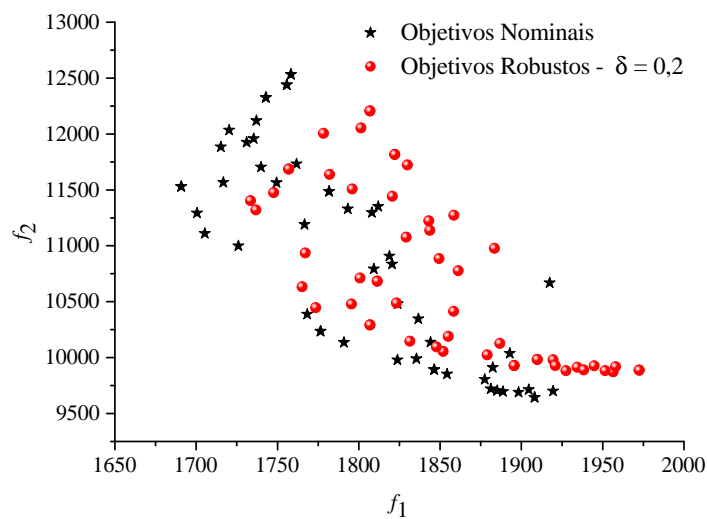


Figura 9.25:  $y_1^{met} \times y_2^{met}$  para o problema do projeto ótimo robusto ( $\delta = 0,2$ ) de um robô industrial.

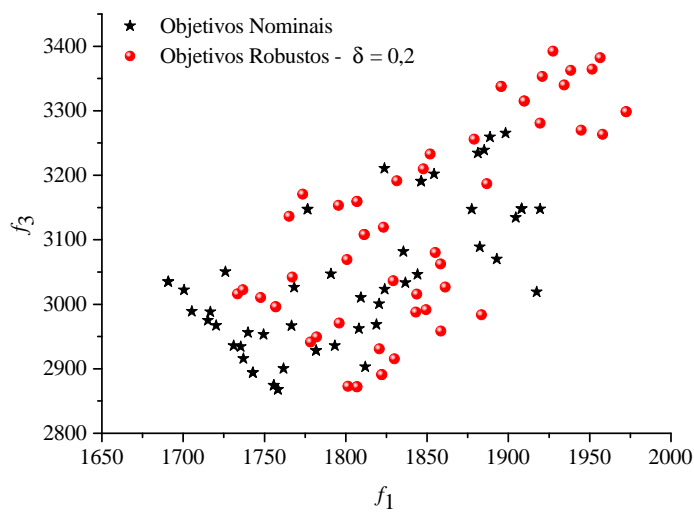


Figura 9.26:  $y_1^{met} \times y_3^{met}$  para o problema do projeto ótimo robusto ( $\delta = 0,2$ ) de um robô industrial.

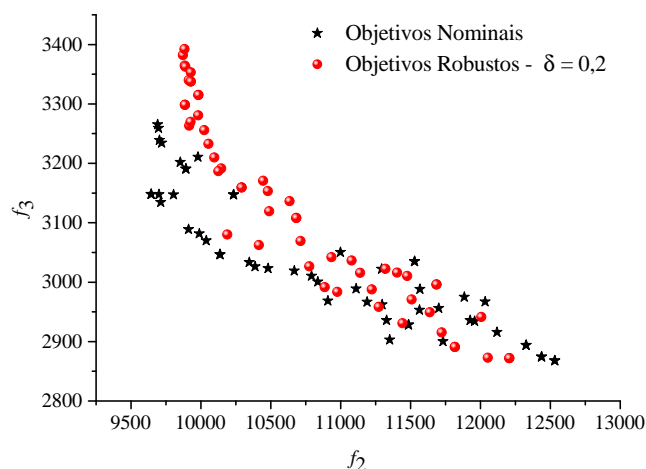


Figura 9.27:  $y_2^{met} \times y_3^{met}$  para o problema do projeto ótimo robusto ( $\delta = 0,2$ ) de um robô industrial.

As Tabelas 9.14, 9.15 e 9.16 apresentam pontos da curva solução nominal e robustas com  $\delta = 0,1$  e  $\delta = 0,2$  respectivamente. São apresentados os valores do objetivo  $y_3^{met}$  extremos, em todos os três casos, ou seja, são apresentados os valores máximo e mínimo obtidos pelo algoritmo MOFA. A solução intermediária é a que apresenta o valor do objetivo  $y_3^{met}$  mais próximo possível à média aritmética dos valores extremos. Observa-se que com o aumento do valor do parâmetro de robustez  $\delta$ , há uma aumento também dos valores dos três objetivos considerados.

Tabela 9.14: Pontos da curva de Pareto nominal para o projeto ótimo de um robo industrial.

Parâmetro	SOLUÇÃO		
$x_1$	-0,687	-0,863	-0,974
$x_2$	0,604	0,346	-0,172
$x_3$	-0,988	-0,935	0,998
$x_4$	-0,964	-0,349	-0,201
$y_1^{met} \times 10^3$	1,936	1,783	1,777
$y_2^{met} \times 10^3$	9,649	10,280	11,866
$y_3^{met} \times 10^3$	3,197	3,033	2,869

Tabela 9.15: Pontos da curva de Pareto robusta com  $\delta = 0,1$  para o projeto ótimo de um robo industrial.

Parâmetro	SOLUÇÃO		
$x_1$	0,866	-0,889	-0,976
$x_2$	0,503	0,339	-0,584
$x_3$	0,503	-0,932	0,995
$x_4$	-0,993	-0,702	-0,019
$y_1^{met} \times 10^3$	1,893	1,834	1,776
$y_2^{met} \times 10^3$	9,763	10,068	12,330
$y_3^{met} \times 10^3$	3,278	3,080	2,868

Tabela 9.16: Pontos da curva de Pareto robusta com  $\delta = 0,2$  para o projeto ótimo de um robo industrial.

Parâmetro	SOLUÇÃO		
$x_1$	0,925	0,313	-0,977
$x_2$	0,003	0,125	-0,735
$x_3$	-0,998	-0,859	0,999
$x_4$	-0,959	-0,447	-0,054
$y_1^{met} \times 10^3$	1,908	1,826	1,801
$y_2^{met} \times 10^3$	9,885	10,588	12,522
$y_3^{met} \times 10^3$	3,361	3,105	2,873

As Figuras 9.28, 9.29, 9.30, 9.31, 9.32 e 9.33 que se seguem, apresentam as relações existentes entre as variáveis de projeto para os casos nominal e robustos com  $\delta = 0,1$  e  $\delta = 0,2$  para o projeto ótimo de um robo industrial. As variáveis de projeto apresentam-se codificadas.

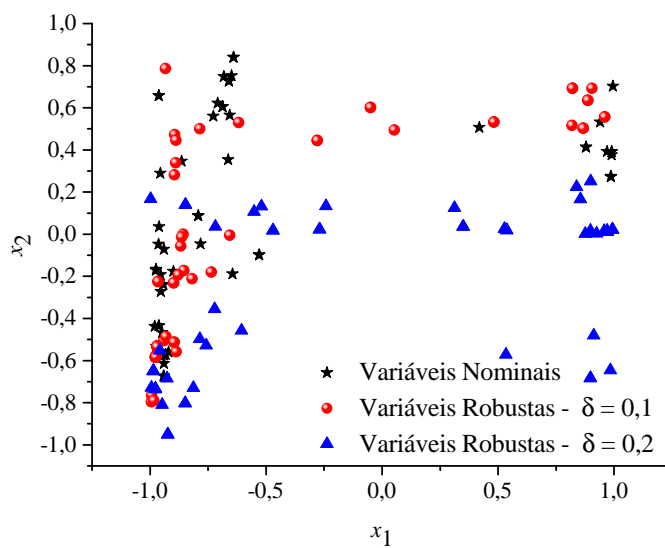


Figura 9.28: Relação  $x_1 \times x_2$  para o problema do projeto ótimo robusto de um robô industrial.

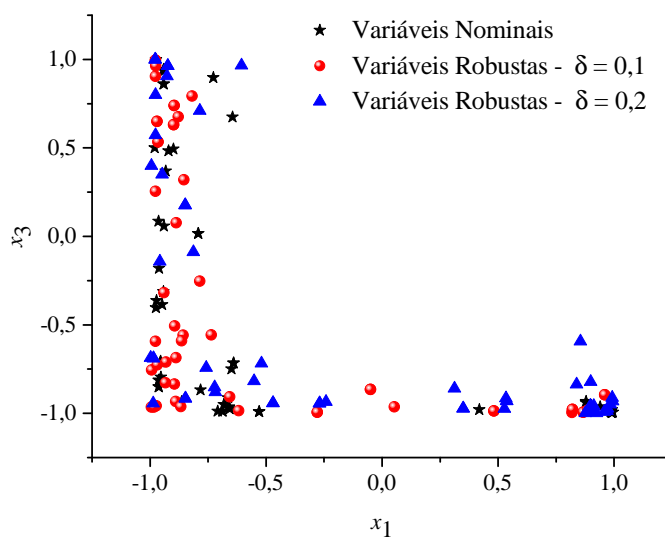


Figura 9.29: Relação  $x_1 \times x_3$  para o problema do projeto ótimo robusto de um robô industrial.

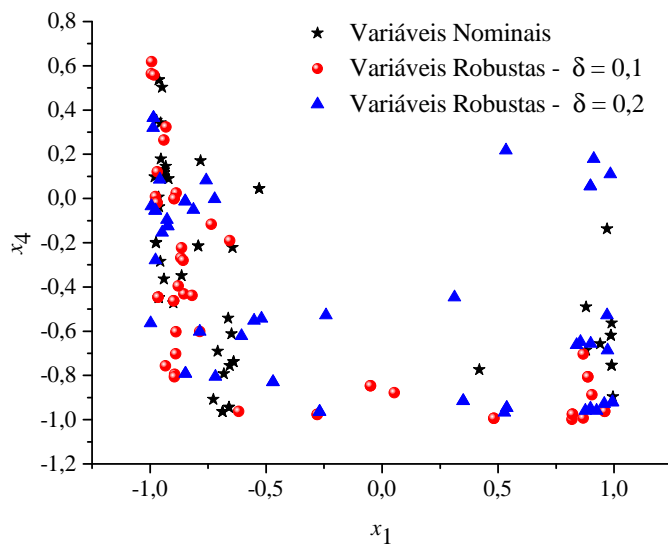


Figura 9.30: Relação  $x_1 \times x_4$  para o problema do projeto ótimo robusto de um robô industrial.

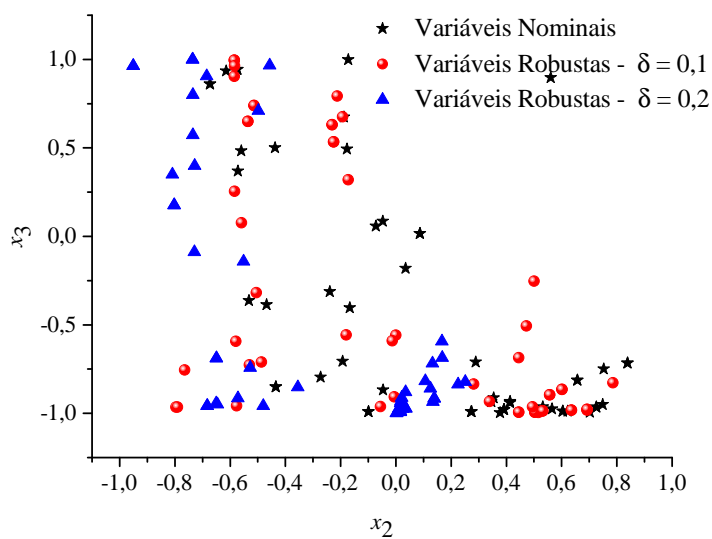


Figura 9.31: Relação  $x_2 \times x_3$  para o problema do projeto ótimo robusto de um robô industrial.



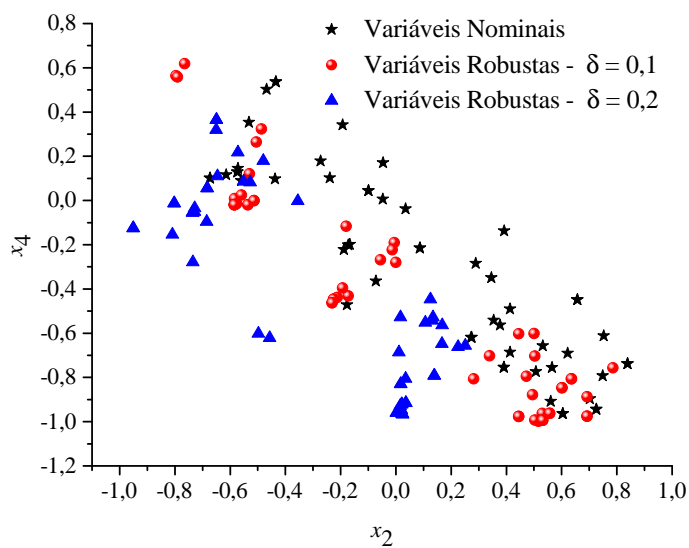


Figura 9.32: Relação  $x_2 \times x_4$  para o problema do projeto ótimo robusto de um robô industrial.

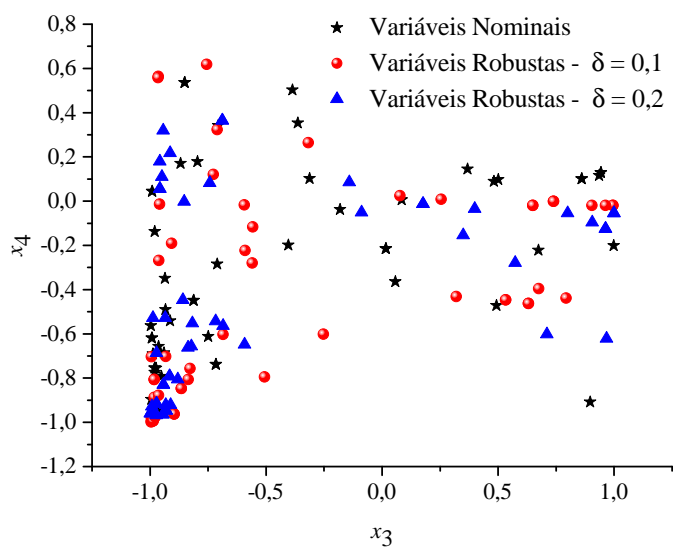


Figura 9.33: Relação  $x_3 \times x_4$  para o problema do projeto ótimo robusto de um robô industrial.

## 9.5 Problema em Dinâmica de Rotores

Na literatura especializada observa-se uma vasta gama de aplicações envolvendo máquinas rotativas, também denominadas rotores, em vários ramos da engenharia, tais como: automotiva, aeroespacial e na geração de energia. Pode-se citar como exemplo de rotores os compressores, máquinas centrífugas, turbinas de avião, turbinas para a produção de energia elétrica, entre outras (TSUZUKI, 2012; CAVALINI, 2013).

Segundo Tsuzuki (2012) o estudo de máquinas rotativas aparece no contexto de máquinas e estruturas devido ao grande número de fenômenos típicos importantes para sua operação e que afetam seu comportamento dinâmico e estrutural. Neste contexto, é de fundamental importância o estudo de máquinas rotativas e a identificação de condições ótimas de operação e de projeto, pois um problema inesperado em uma máquina rotativa pode conduzir a consideráveis riscos e perdas financeiras. Por exemplo, Quitzrau (2002) afirma que é muito comum a adoção de configurações com mais mancais que o necessário ou, eixos mais rígidos, para aumentar os valores das frequências críticas, elevando os custos com mão de obra e de fabricação, além de elevar perdas por atrito no sistema quando o número de mancais é maior que o necessário.

Devido a grande utilização de máquinas rotativas como componentes em diversas estruturas, faz-se necessário determinar condições ótimas de operação bem como de projeto dos rotores. Sendo assim, o uso de técnicas de otimização para o projeto ótimo de máquinas rotativas representa uma importante abordagem alternativa.

Na literatura corrente vários trabalhos têm sido apresentados para o projeto e identificação de parâmetros em máquinas rotativas. Assis (1999) utiliza técnicas de otimização para o projeto e identificação de parâmetros em máquinas rotativas. Steffen Jr., Assis e Lepore Neto (1999) utilizaram técnicas multiobjetivo para a otimização de rotores. He,

Guo e Chu (2001) utilizam a heurística Algoritmos Genéticos, associado ao método de elementos finitos, para detectar trincas em eixos de rotores. Assis e Steffen Jr. (2003) utilizam técnicas de otimização, usando problemas inversos, para identificação de parâmetros em máquinas rotativas. Saldarriaga et al. (2011) utilizaram a abordagem de problemas inversos para estudar o balanceamento de máquinas rotativas flexíveis. Bueno (2007) utilizou técnicas de otimização para obtenção de localizações ótimas de sensores e atuadores piezelétricos para o controle ativo de vibrações. Tsuzuki (2012) apresenta, em um capítulo do livro editado por ele, a otimização do eixo de uma máquina rotativa usando o método recozimento simulado. Cavalini Jr. (2013), em sua tese de doutorado, utiliza a heurística Evolução Diferencial para identificar e caracterizar trincas transversais incipientes em eixos de rotores.

### **9.5.1 Otimização do Eixo de um Rotor**

A velocidade crítica é o ponto onde a velocidade de rotação se iguala à frequência natural do rotor. Sabe-se que nas velocidades críticas as amplitudes de vibração de um sistema rotativo são máximas e se uma máquina rotativa operar em velocidade próxima à uma velocidade crítica o sistema pode entrar em ressonância, podendo ocasionar danos irreversíveis aos componentes do sistema rotativo. Sendo assim, é importante que um rotor opere em uma velocidade de rotação distante das velocidades críticas. No contexto de robustez é importante conhecer as regiões de projeto onde há pouca sensibilidade das velocidades críticas devido a pequenas perturbações das variáveis de projeto. Um diagrama de Campbell, apresentado na Fig. 9.34 a seguir, mostra duas velocidades críticas de um sistema rotativo e ilustra a amplitude de vibração do mesmo. Ao diagrama de Campbell foi sobreposta uma curva de resposta em frequência, para fins de ilustração.

Neste contexto, a presente aplicação objetiva a maximização da distância entre a

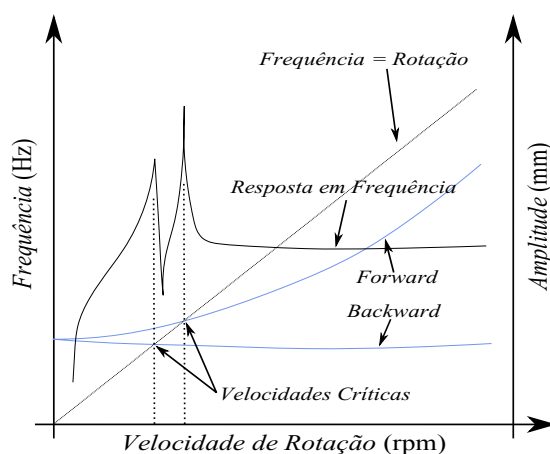


Figura 9.34: Diagrama de Campbell genérico.

terceira e a segunda velocidades críticas (em *forward*) para um sistema rotativo, modelado pelo método de elementos finitos, composto pelo eixo do rotor, por três discos rígidos, dois mancais flexíveis e isotrópicos, ou seja, com amortecimento e rigidez e os termos de acoplamento nulos, respectivamente, conforme mostra a Fig. 9.35. Assis (1999) apresenta em detalhes a formulação do modelo de elementos finitos utilizada neste estudo de caso.

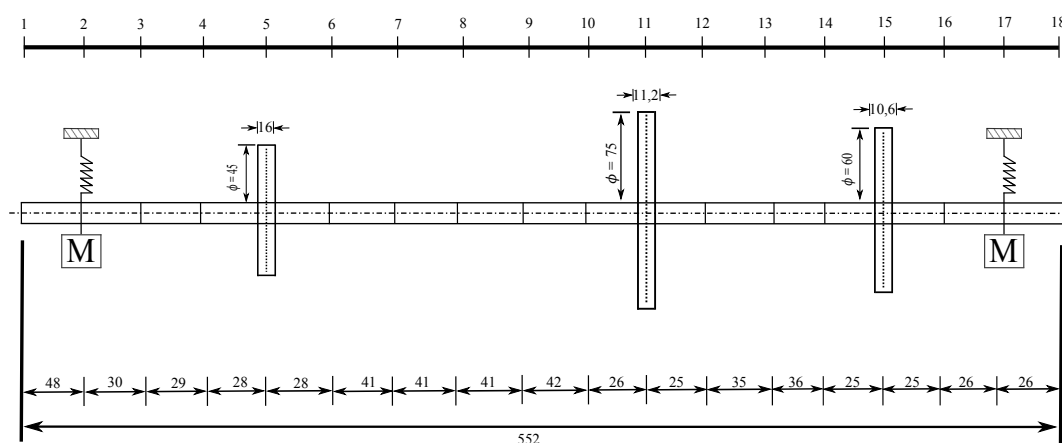


Figura 9.35: Modelo de elementos finitos para o rotor estudado (ASSIS, 1999).

O material utilizado no eixo e nos discos é o aço-1020 (densidade -  $7800 \text{ kg/m}^3$ , módulo de elasticidade -  $2.1\text{E}11 \text{ N/m}^2$  e coeficiente de Poisson - 0.3). As características

geométricas dos discos são apresentadas na Tab. 9.17 a seguir:

Tabela 9.17: Características geométricas dos discos da máquina rotativa.

Disco	Massa (Kg)	Momento de Inércia (Kg m <sup>2</sup> )	Raio (mm)	Espessura (mm)
1	0,794	0,005	45	16
2	1,544	0,005	75	11,2
3	0,935	0,005	60	10,6

Os mancais de rolamentos são idênticos com massa de 10 Kg, rigidezes  $K_{xx} = K_{zz} = 1E6N/m$ ,  $K_{xz} = K_{zx} = 0$  com coeficientes de amortecimentos  $C_{xx} = C_{zz} = 83,2615Ns/m$  e  $C_{xz} = C_{zx} = 0$ .

Para a maximização da distância entre a terceira e a segunda velocidades críticas, onde as variáveis de projeto consideradas são os raios dos dezessete elementos em que o eixo do rotor foi discretizado pelo método de elementos finitos, é formulado o seguinte problema de otimização multiobjetivo:

$$\text{minimizar } f_1(x_1, x_2, \dots, x_{17}) = V_c(2) \quad (9.20)$$

$$\text{maximizar } f_2(x_1, x_2, \dots, x_{17}) = V_c(3) \quad (9.21)$$

onde  $V_c$  é o vetor de velocidades críticas,  $x_i =$  raio do  $i$ -ésimo elemento de barra,  $i = 1, \dots, 17$  com  $4 \text{ mm} \leq x_i \leq 8 \text{ mm}$ ,  $i = 1, \dots, 17$ .

Para a avaliação da influência do parâmetro de robustez no processo de otimização, foi utilizado o seguinte vetor de desvios  $[\delta_1 \dots \delta_{17}] = [\delta \dots \delta]$ , para os seguintes valores do parâmetro de perturbação  $\delta = 0,0002$ ,  $\delta = 0,0004$  e  $\delta = 0,0006$ , que representam uma perturbação de 5%, 10% e 15% respectivamente, da amplitude de variação das variáveis de projeto.

Os parâmetros utilizados no algoritmo MOFA durante as simulações foram:

- tamanho da população -  $N_{vag} = 35$  vagalumes;
- número de gerações -  $N_{ger} = 250$  (utilizado como critério de parada);
- número de amostras para calcular a média efetiva -  $N_{amt} = 50$ ;
- coeficientes de absorção da luz e de atratividade entre os vagalumes -  $\gamma = \beta_0 = 1$ .

A Figura 9.36, apresenta as curvas de Pareto nominal e robustas obtidas pelo algoritmo MOFA para os valores do parâmetro de perturbação acima citados.

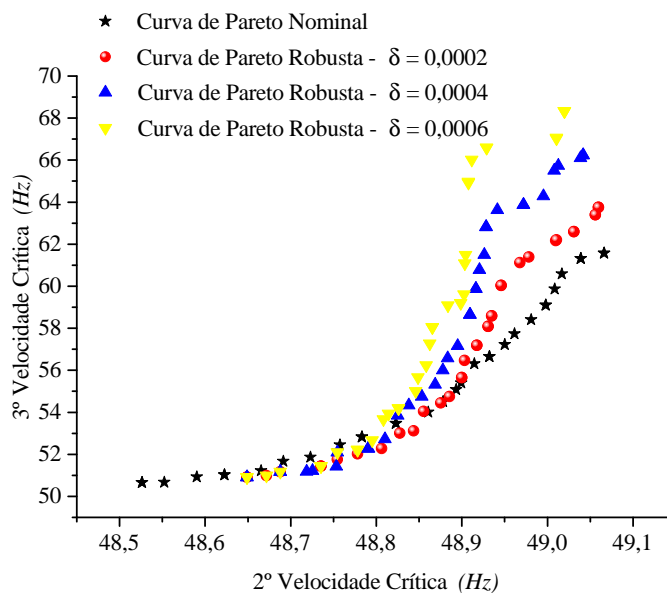


Figura 9.36: Curvas de Pareto nominal e robustas obtidas pelo algoritmo MOFA para o problema de uma máquina rotativa.

Observa-se nesta figura que, para valores das variáveis de projeto que fazem com que a segunda velocidade crítica seja maior que 48,9 Hz, há uma grande sensibilidade do segundo objetivo, ou seja, da terceira velocidade crítica para pequenas perturbações nas variáveis de projeto. Neste caso, é importante que a escolha das variáveis de projeto satisfaçam a condição de que a segunda velocidade crítica esteja entre 48,6 Hz e 48,9 Hz,

pois nesta faixa o segundo objetivo, ou seja, a terceira velocidade crítica é pouco sensível a pequenas perturbações no projeto. Então pode-se considerar a faixa citada como uma região de robustez para o projeto ótimo do eixo da máquina rotativa estudada.

Tabela 9.18: Máxima distância entre as velocidades críticas para os casos robustos e nominal.

Variável	Valor Médio	Nominal	Robusta ( $\delta_a$ )	Robusta ( $\delta_b$ )	Robusta ( $\delta_c$ )
$x_1$	6,00	6,10	5,78	5,47	5,96
$x_2$	6,00	5,11	6,15	6,16	6,13
$x_3$	6,00	7,80	6,01	7,27	6,34
$x_4$	6,00	7,88	7,25	7,27	7,64
$x_5$	6,00	7,88	6,21	6,27	6,10
$x_6$	6,00	7,94	7,97	7,98	7,99
$x_7$	6,00	7,99	7,51	7,54	7,49
$x_8$	6,00	7,90	7,63	7,84	7,65
$x_9$	6,00	7,74	7,86	7,78	7,95
$x_{10}$	6,00	7,95	7,96	7,99	7,97
$x_{11}$	6,00	6,70	7,98	7,83	7,99
$x_{12}$	6,00	7,74	7,53	7,54	7,62
$x_{13}$	6,00	7,85	7,92	7,92	7,90
$x_{14}$	6,00	7,24	7,57	7,65	7,96
$x_{15}$	6,00	7,45	6,77	7,02	6,73
$x_{16}$	6,00	6,51	7,39	7,39	7,57
$x_{17}$	6,00	5,71	7,15	7,15	7,15
$V_c(2)$	49,064	49,066	49,059	49,042	49,020
$V_c(3)$	52,573	61,573	63,755	66,238	68,335
$V_c(3) - V_c(2)$	3,509	12,506	14,695	17,196	19,315

onde  $\delta_a = 0,0002$ ,  $\delta_b = 0,0004$  e  $\delta_c = 0,0006$ , as variáveis de projeto são expressas em milímetros e as velocidades críticas em Hertz.

É importante ressaltar que a implementação de um eixo com os valores das variáveis, para o casos robustos, não implica que a diferença entre as velocidades críticas seja 14,695, 17,196 e 19,315. O que ocorre é, se for permitida uma tolerância de projeto com os valores de perturbação  $\delta_a$ ,  $\delta_b$  e  $\delta_c$ , então os valores máximos das distâncias entre as velocidades críticas, nos intervalos de definição das variáveis perturbadas  $x_i \pm \delta_a$ ,  $x_i \pm \delta_b$  e  $x_i \pm \delta_c$ ,  $i = 1, \dots, 17$ , são 14,695, 17,196 e 19,315 respectivamente.

As Figuras 9.37, 9.38 e 9.39 ilustram os diagramas de Campbell considerando as

variáveis de projeto apresentadas na segunda, terceira e sexta coluna da Tab. 9.18.

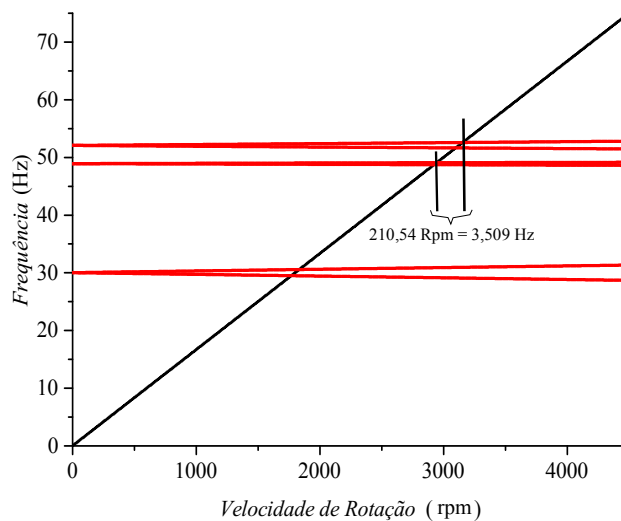


Figura 9.37: Diagrama de Campbell para o valor médio apresentado na Tab. 9.18.

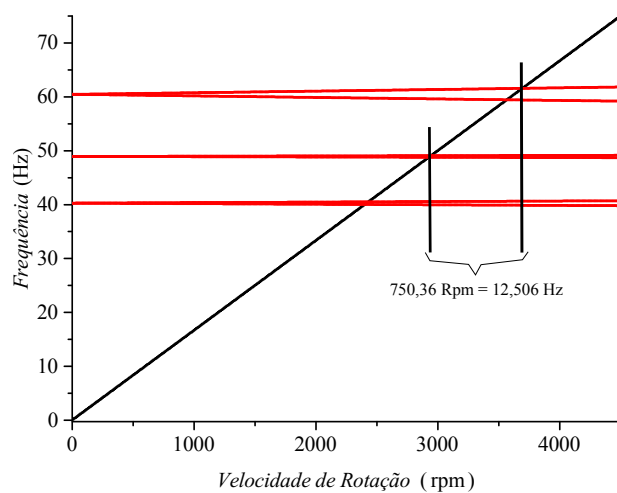


Figura 9.38: Diagrama de Campbell para o caso nominal apresentado na Tab. 9.18.



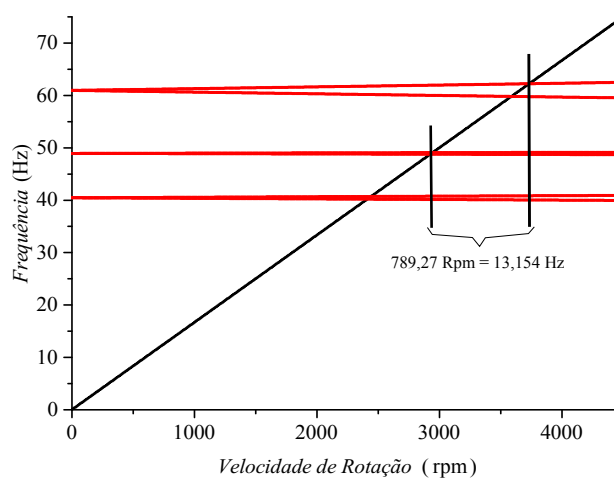


Figura 9.39: Diagrama de Campbell para o caso robusto com  $\delta = 0,0006$ .

A Figura 9.40 ilustra os três casos anteriores, agora apresentados em um único diagrama de Campbell (apenas em *forward*).

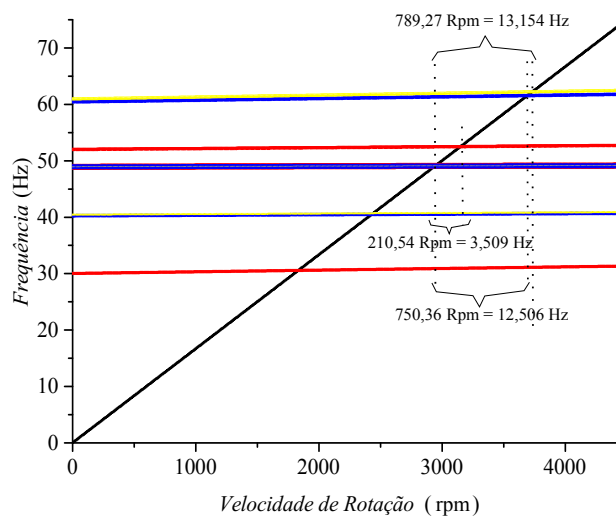


Figura 9.40: Diagrama de Campbell para os três casos estudados.

## 9.6 Conclusões Preliminares

Este capítulo foi dedicado à resolução de alguns problemas em engenharia. Os problemas abrangeram o estudo de projetos estruturais, uso de metamodelagem no estudo de um robô industrial e uma aplicação em dinâmica de máquinas rotativas. Visto que não há um estudo sistemático introduzindo robustez em problemas multiobjetivo de otimização, os problemas estudados servirão como objeto de comparação para a avaliação de outras metodologias de otimização multiobjetivo robustas a serem desenvolvidas. Como avaliação dos resultados obtidos nos problemas abordados, pode-se concluir que a metodologia proposta para o tratamento de problemas de otimização multiobjetivo robustos, com base no emergente heurística de Colônia de Vagalumes e associada ao conceito de média efetiva para inserção de robustez ao problema, trata-se de uma metodologia viável para resolução de problemas reais em engenharia.

# Capítulo 10

## Considerações Finais e Sugestões

Neste capítulo são apresentadas as principais conclusões e contribuições acerca do algoritmo desenvolvido, bem como as sugestões de trabalhos futuros.

### 10.1 Conclusões

Os algoritmos baseados em processos evolutivos têm sido largamente utilizados pela comunidade acadêmica em geral para resolver os mais diversos problemas de otimização. Porém, nos últimos anos, o número de trabalhos e algoritmos baseados no comportamento social de pássaros, abelhas, formigas, aves, peixes, vagalumes, entre outros, tem aumentado consideravelmente. A vantagem da utilização das heurísticas se devem à capacidade que estas possuem em tratar problemas reais e a habilidade destas de escapar de soluções ótimas locais. Porém, talvez o fator mais importante da ampla utilização destas heurísticas se deve ao crescente aprimoramento das ferramentas computacionais e da disponibilização destes recursos no meio acadêmico. Neste contexto, foi desenvolvido neste trabalho o algoritmo MOFA (*Multiobjective Optimization Firefly Algorithm*), funda-

---

mentado na extensão do algoritmo Colônia de Vagalumes, proposto por Yang (2008) para resolução de problemas mono-objetivo, para o tratamento de problemas multiobjetivo e associado ao conceito de média efetiva, proposto por Deb e Gupta (2006) para incorporar incertezas associadas à produção de tolerâncias às variáveis de projeto e parâmetros dos objetivos. Os principais mecanismos de extensão do algoritmo para o caso multiobjetivo são: ordenamento das curvas de Pareto por meio de *rank* e o truncamento de soluções por meio do conceito de distância de multidão (DEB, 2001).

Para avaliar a eficiência do algoritmo MOFA foram consideradas duas métricas, a saber: convergência métrica e a diversidade de soluções. Estas métricas são conhecidas como: Diversidade Métrica e Convergência Métrica (DEB, 2001). Para avaliar as métricas é necessário conhecer a curva de Pareto real do problema. No contexto robusto, para avaliar a convergência do algoritmo faz-se necessário também conhecer as curvas de Pareto robustas reais do problema. Foram estudados os problemas propostos por Deb e Gupta (2006), onde as médias efetivas foram obtidas analiticamente através de sua definição para problemas com e sem restrições. Em se tratando de convergência e diversidade, verificou-se a eficiência do algoritmo MOFA, pois houve a convergência para as curvas de Pareto nominais e robustas para vários valores do parâmetro de perturbação  $\delta$  e também houve uma boa distribuição das soluções sobre as curvas de Pareto nominais e robustas. Foram estudados alguns outros problemas matemáticos, porém sem curvas de Pareto robustas conhecidas, objetivando avaliar o efeito do parâmetro de perturbação sobre as curvas robustas em relação à curva de Pareto nominal, ou seja, para localizar as zonas de robustez dos problemas estudados. O algoritmo MOFA mostrou-se capaz de convergir para a curva de Pareto nominal e de assegurar uma boa diversidade das soluções.

Na literatura há uma proposta de que os parâmetros  $\beta_0$  e  $\gamma$  do algoritmo ACV variem nos intervalos  $[0, 1]$  e  $[0, 10]$ , respectivamente (YANG, 2008). No contexto robusto o

---

parâmetro tamanho da amostra  $N_{\text{amt}}$  foi considerado para estimar a média efetiva. Estudando a sensibilidade destes parâmetros em relação a convergência e a diversidade das soluções, concluiu-se que os melhores valores, pelo menos para o estudo de caso analisado, foram  $\beta_0 = \gamma = 1$  e que o tamanho da amostra deve ser de no mínimo 50 vagalumes para se obter uma boa qualidade na solução. Outras estratégias para estimar a integral devem ser avaliadas para tentar obter um número menor no tamanho da amostra para estimar a média efetiva, visto que este número onera muito todo processo computacional.

Objetivando aplicar a metodologia proposta em problemas de engenharia foram estudados cinco situações envolvendo estruturas, robô industrial cujo comportamento é definido a partir de técnicas de metamodelagem, e dinâmica de máquinas rotativas. Em todas as situações realísticas abordadas o algoritmo MOFA mostrou-se bastante eficiente na convergência (os quatro primeiros problemas são encontrados na literatura corrente), diversidade de soluções (observadas graficamente) e também na verificação da influência do parâmetro de robustez sobre as curvas de Pareto e sobre as variáveis de projeto.

É importante ressaltar que os operadores utilizados no algoritmo MOFA não foram desenvolvidos neste trabalho. São operadores que foram exaustivamente testados e aprovados na literatura corrente. O conceito de Média Efetiva também não foi proposto originalmente nesta tese. Contudo, o acoplamento dos operadores e do conceito de Média Efetiva associado ao Algoritmo de Colônia de Vagalumes configura uma nova abordagem metodológica, pelo menos no que diz respeito àquelas que foram investigadas na literatura. Além disso, não há um número significativo de trabalhos considerando o uso do Algoritmo Colônia de Vagalumes em problemas multiobjetivo e menos ainda algoritmos multiobjetivo considerando a inserção de robustez ao processo de otimização. Sendo assim, o desenvolvimento de um algoritmo multiobjetivo para otimização robusta usando uma heurística emergente, não baseada em processos evolutivos, se evidencia como a

principal contribuição deste trabalho.

### **10.1.1 Contribuições**

As principais contribuições desta tese foram:

- Organização e disponibilização de material na literatura de tratamento de problemas de otimização multiobjetivo robustos, exibindo as principais estratégias de incorporação de incertezas ao problema e enfatizando a metodologia baseada no conceito de média efetiva;
- Desenvolvimento de um código computacional, em ambiente Matlab<sup>®</sup>, com base no Algoritmo Colônia de Vagalumes associado ao conceito de média efetiva para a inserção de robustez ao problema;
- Resolução de diversos problemas matemáticos clássicos com solução conhecida e de problemas realísticos em engenharia, objetivando tanto a comprovação da eficácia do algoritmo proposto quanto a formação de banco dados para futuras comparações com outras estratégias de otimização multiobjetivo robusta.

### **10.1.2 Sugestões para Trabalhos Futuros**

Como sugestão para estudos futuros relacionados ao presente trabalho, pode-se citar:

- Desenvolver o algoritmo MOFA em ambiente de programação C, permitindo assim um ganho no tempo de processamento, visto que o número de avaliações dos objetivos para o caso robusto é consideravelmente maior do que no caso nominal;
- Aplicar a metodologia proposta em várias outros problemas reais em engenharia;

- Comparar estratégias de aproximação da integral da definição de média efetiva, tanto em relação ao custo computacional quanto a qualidade da curva de Pareto obtida;
- Estudar incertezas associadas à metamodelagem;
- Utilizar dados experimentais reais para a geração dos metamodelos, verificando em seguida a eficiência da metodologia proposta na resolução dos problemas inversos.

# Referências Bibliográficas

ABBASS, H. A.; SARKER, R.; NEWTON, C. Pde: A Pareto-frontier differential evolution approach for multi-objective optimization problems. *Proceedings of the 2001 Congress on Evolutionary Computational*. USA: IEEE, 2001.

AGARWAL, H., *Reliability Based Design Optimization: Formulation and Metodologies*. Thesis (Ph.d.). University of Notre Dame, South Bend, IN, 2004.

ANGIRA, R., BABU, B. V., Non-dominated sorting differential evolution(NSDE): An extension of differential evolution for multiobjective optimization. *Proceedings of the 2nd Indian International Conference on Artificial Intelligence*. India: II CAI-2005, 2005.

ASSIS, E. G., *Uso de Técnicas de Otimização para Auxiliar o Projeto e Identificar parâmetros em Máquinas rotativas*. Tese (Doutorado). Faculdade de Engenharia Mecânica. Universidade Federal de Uberlândia. 1999.

ASSIS, E.G., STEFFEN Jr., V., Inverse Problems Techniques for the Identification of rotor-bearing systems. *Inverse Problems in Engineering*, v.11, nº 1, p. 39-53, 2003.

ATHAN, T. W., PAPALAMBROS, P. Y. A note on weighted criteria methods for compromise solutions in multi-objective optimization. *Eng. Optimization*, v. 27, n. 3, p.155-176, 1996.

BABU, B. V., ANBARASU, B., Muti-objective differential evolution (MODE):



---

An evolutionary algorithm for multi-objective optimization problems (MO-OPS). *Proceedings of International Symposium and 58th Annual Session of IChE*. Indian, CHEMCON-2005, 2005.

BABU, B. V., CHAKOLE, P. G., MUBEEN, J. H. S. Multiobjective Differential Evolution (MODE) for optimization of adiabatic styrene reactor, *Chemical Engineering Science*, Vol. 60, p.4822-4837, 2005.

BARROS, M.B., *Proposição, Avaliação Numérica e Experimental de um Absorvedor Dinâmico de Vibrações Multimodal*. Dissertação (Mestrado). Faculdade de Engenharia Mecânica. Universidade Federal de Uberlândia. 2009.

BAUER, W.F., The Monte Carlo Method. *Journal of Society for Industrial and Applied Mathematics*, Vol. 06, nº 04, p. 438-451, 1958.

BENNET, A.A., Newton's Method in General Analysis. *Proc. Nat. Acad. Sci. USA* 2(10), p.592-598, 1916.

BENNET, J. A., LUST, R. V., Conservative Methods for Structural Optimization. *AIAA Journal*. Vol. 28, p. 1491-1496, 1990.

BERTSEKAS, D.P. *Nonlinear Programming*, Segunda Edição. Athena, Belmont, 1999.

BEYER, H.-G., SENDHOFF, B., *Robust Optimization - A Comprehensive Survey*. Elsevier Science. 2007.

BORGES, R. A. *Contribuição ao Estudo de Absorvedores Dinâmicos de Vibrações Não-Lineares*. Tese(Doutorado). Faculdade de Engenharia Mecânica. Universidade Federal de Uberlândia. 2008.

BORGES, R.A., LIMA, A.M.G., STEFFEN Jr., V., Robust Optimal Design of a Nonlinear Dynamic Vibration Absorber Combining Sensitivity Analysis. *Shock and Vibration*. V. 17, p. 507-520, 2010.

---

BORGES, R.A., LOBATO, F.S., STEFFEN Jr., V., Application of Three Bioinspired Optimization Methods for the Design of a Nonlinear Mechanical System. *Mathematical Problems in Engineering*. 2013.

BOWMAN, V.J. Jr. On the Relationship of the Tchebycheff Norm and the Efficient Frontier of the Multi-Criteria Objectives. Multiple Criteria Decision Making, Edited by H. Thiriez, S. Zionts, *Lecture Notes in Economics and Mathematical Systems*, 130, Springer-Verlag, Berlin, p.65-76, 1976.

BOX, G. E. P.; HUNTER, W. G.; HUNTER, J. S., Statistics for Experiments - an Introduction to Design, Data Analysis and Model Building. First edition. New York: John Wiley & Sons, 1978.

BOYCHUK, L. M.; OVCHINNIKOV, V. O. Principal methods of solution multi-criterial optimization problems. *Soviet Automatic Control*, v. 6, p.1-4, 1973.

BOYER, C.B., MERZBACH, U.C., *A History of Mathematics*. 3<sup>o</sup> Edition. John Wiley & Sons, Inc. 2011.

BRANDÃO, M.A.L. *Estudo de Alguns Métodos Determinísticos de Otimização Irrestrita*. Dissertação Mestrado. Faculdade de Matemática. Universidade Federal de Uberlândia. 2010.

BRANKE, J., *Evolutionary Optimization in Dynamic Environments*. Volume 3. Springer EUA, 2002.

BRANKE, J., DEB, K., MIETTINEN, K., SLOWINSKI, R. *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Springer-Verlag. Berlin, 2008.

BRIDGMAN, P. W. *Dimensional Analysis*. First edition. New Haven: Yale University Press, 1922.

BUENO, D. D., *Controle Ativo de Vibrações e Localização Ótima de Sensores e Atuadores Piezelétricos*. Dissertação (Mestrado). Faculdade de Engenharia de Ilha

---

Solteira, Universidade Estadual Paulista, 2007.

BURDEN, R.L., FAIRES, J.D. *Numerical Analysis*. 8th edition. Thomson Learning, 2005.

CABALERO, R., REY, L., RUIZ, F., GONZÁLEZ, M. An Algorithm Package for the Resolution and Analysis of Convex Multiple Objective Problems. Multiple Criteria Decision Making: Proceedings of the Twelfth International Conference, Hagen (Germany). Edited by G. Fandel, T. Gal, *Lecture Notes in Economics and Mathematical Systems 448*, Springer Verlag, Berlin, Heidelberg, p. 275-284, 1997.

CARTIS, C., GOULD, N.I.M., TOINT, Ph.L. On the Complexity of Steepest Descent, Newton's and Regularized Newton's Methods for Nonconvex Unconstrained Optimization Problems. *SIAM Journal on Optimization*, Vol. 20, Issue 6, p. 2833-2852, 2010.

CASTRO, R. E. *Otimização de Estruturas com Multi-Objetivos via Algoritmos Genéticos*. Tese (Doutorado), Departamento de Engenharia Civil - COPPE/UFRJ, 2001.

CAVALINI, A. A. Jr., *Detecção e Identificação de Trincas Transversais Incipientes em Eixos Horizontais Flexíveis de Máquinas Rotativas*. Tese (Doutorado). Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, 2013.

CHANKONG, V., HAIMES, Y.Y. Optimization-Based Methods for Multiobjective Decision-Making: An Overview. *Large Scale Systems*, 5, n°1, p.1-33, 1983a.

CHANKONG, V., HAIMES, Y.Y. *Multiobjective Decision Making Theory and Methodology*. Elsevier Science Publishing, Co. Inc. New York, 1983b.

CHEN, W., ALLEN, J.K., TSUI, K.-L., MISTREE, F., A Procedure for Robust Design: Minimizing Variations Caused by Noise Factors and Control Factors. *ASME Journal of Mechanical Design*, (118): p.478-485, 1996.

- 
- COELHO, R. F. *Multi-Criteria Optimization With Expert Rules for Mechanical Design*. Tese (Doutorado). Université Libre de Bruxelles, Faculté des Sciences Appliquées, 2004.
- CORLEY, H.W. A New Scalar Equivalence for Pareto Optimization. *IEEE Transactions on Automatic Control*, 25, n°04, p.829-830, 1980.
- DAS, I., DENNIS, J. E. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural Optimization*, v. 14, n. 4, p. 63-69, 1997.
- DAS, I., Robustness Optimization for Constrained Nonlinear Programming Problems. *Engineering Optimization*, 32(5), p. 585-618, 2000.
- DEB, K. *Multi-objective Optimization Using Evolutionary Algorithms*. First Edition. New York: John Wiley & Sons, 2001.
- DEB, K.; AGRAWAL, S.; PRATAB, A.; MEYARIVAN, T. *A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II*. Kanpur - India, 2000.
- DEB, K., GUPTA, H., Introducing Robustness in Multi-Objective Optimization. *Evolutionary Computation*. 14(4):p. 463-494, 2006.
- DOERNER, K., GUTJAHR, W.J., HARTL, R.F., STRAUSS, C., STUMMER, C. Ant Colony Optimization in Multiobjective Portfolio. In: *Proceedings of the 4th Metaheuristics International Conference MIC'2001*. USA: Souza J.P. (ed), p.243-248, 2001.
- DOMINGUES, H. H., IEZZI, G. *Álgebra Moderna*. 4º Edição. Editora Atual. São Paulo, 2003.
- DU, X., WANG, Y., CHEN, W., Methods for Robust multidisciplinary Design. *American Institute of Aeronautics and Astronautics AIAA*. 2000.

- 
- DUNLAP, R. A. *The Golden Ratio and Fibonacci Numbers*. World Scientific Publishing, 1997.
- EDGAR, T. F.; HIMMELBLAU, D. M.; LASDON, L. S. *Optimization of Chemical Process*. Second edition. New York: McGraw-Hill Chemical Engineering Series, 2001.
- EDGEWORTH, F.Y. *Mathematical Physics*. First Edition. London, England: P. Keagan, 1881.
- EGLAJS, V., AUDZE, P., New approach to the design of multifactor experiments. *Problems of Dynamics and Strengths*. 35 (in Russian) (Riga: Zinatne Publishing House), p.104-107, 1977.
- EL GHAOUI, L. LEBRET, H., Robust Solutions to least-squares problems with uncertain data. *SIAM J. Matrix Analysis and Applications*. 18 (4), p. 1035-1064, 1997.
- ENCYCLOPEDIA BRITANNICA, Firefly. *In: Encyclopedia Britannica*. Ultimate Reference Suite. Chicago: Encyclopedia Britannica, 2009.
- ESCHENAUER, H., KOSKI, J., OSYCZKA, A., *Multicriteria Design Optimization - Procedures and Applications*. First Edition. Berlin Heidelberg: Springer-Verlag, 1990.
- FANG, H., RAIS-ROHANI, M., LUI, Z., HORSTEMEYER, M.F. A Comparative Study of Metamodeling Methods for Multiobjective Crashworthiness Optimization. *Computers and Structures*, v.83, p. 2121-2136, 2005.
- FILHO, J.L.R., TRELEAVEN, P.C., ALIPPI, C. Genetic-Algorithm Programming Environments. *IEEE Computer*, Vol.27, n.6, p.28-43, 1994.
- FINE, H., On Newton's Method of Approximation. *Proc. Nat. Acad. Sci. USA* 2(10), p.546-552, 1916.
- FLETCHER, R., POWELL, M.J.D. A Rapidly Convergent Descent Method for Minimization. *The Computer Journal*, Vol. 06, Issue 2, p. 163-168, 1963.
- FONSECA, C. M., FLEMING, P. J. Genetic algorithms for multiobjective opti-

---

mization: Formulation, discussion and generalization. In: *Proceedings of the Fifth International Conference on Genetic Algorithms*. USA: Gen93, 1993.

FONSECA, C. M., FLEMING, P. J. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation Journal*, v. 3, n. 1, p. 1-16, 1995.

FRANCO, N. B. *Cálculo Numérico*. Editora: Pearson PrenticeHall, São Paulo, 2007.

FRISSE, C., SCARPEL, R. A., FERRARI, D. B. T. P. do A., *Metamodelagem de Funções Determinísticas por Composição Integrada de Especialistas Locais*. XLIII Congresso Brasileiro de Pesquisa Operacional. Ubatuba - SP, 2011.

GANDIBLEUX, X., MEZDAOUI, N., FREVILLE, A. A Tabu Search Procedure to Solve Multiobjective Combinatorial Optimization Procedures. In: *Advances in Multiple Objective and Goal Programming, Lecture Notes in Economics and Mathematical Systems 455, 291-300*. Springer-Verlag: Cabalero R. Ruiz F. and Steuer R. (eds), 1997.

GOLDBERG, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. First Edition. USA: Adison-Wesley, 1989.

HAIMES, Y. Y., LASDON, L.S., WISMER, D.A., On a Bicriterion Formulation of the Problems of the integrated System Identification and System Optimization. *IEEE Transactions on Systems, Man, and Cybernetics 1*, p. 296-297, 1971.

HAIMES, Y. Y.; HALL, W. A. Multiobjective in water resources systems analysis: The surrogate worth trade-off method. *Wat. Resources Res.*, v. 10, p. 615-624, 1974.

HAUPT, R. L.; HAUPT, S. E. *Practical Genetic Algorithms*. First edition. INC.: John Wiley & Sons, 1998.

HE, Y., GUO, D., CHU, F., Using Genetic Algorithms and Finite Elements Methods to detect Shaft Crack for Rotor-Bearing System. *Mathematics and Computers in*

---

*Simulation*, v. 57, n. 1, p.95-108, 2001.

HOLLAND, J.H. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.

HORN, J. *Handbook of Evolutionary Computation*. v. 1, Oxford University Press. Oxford, 1997.

HORN, J.; NAFPLIOTIS, N.; GOLDBERG, D. E. A niched pareto genetic algorithm for multiobjective optimization. In: *Proceedings of the First IEEE Conference on Evolutionary Computation*. Florida - USA: IEEE World Congress on Computational Intelligence, 1994.

IZMAILOV, A., SOLODOV, M. *Otimização: Condições de Otimalidade, Elementos de Análise Convexa e de Dualidade*. Volume 1. Associação Instituto Nacional de Matemática Pura e Aplicada - IMPA. Rio de Janeiro, 2005.

IZMAILOV, A., SOLODOV, M. *Otimização: Métodos Computacionais*. Volume 2. Associação Instituto Nacional de Matemática Pura e Aplicada - IMPA. Rio de Janeiro, 2007.

JIN, Y., SENDHOFF, B., Trade-Off Between Performance and Robustness: An Evolutionary Multiobjective Approach. *Evolutionary Multi-Criterion Optimization*. Lecture Notes in Computer Science. Volume 2632/2003, 2003.

KANTOROVICH, L. V., On Newton's Method for functional Equations. *Dokl. Akad. Nauk SSSR*. 59(7), p.1237-1240, 1948.

KANTOROVICH, L.V., and AKILOV, G.P., *Functional analysis in normed spaces*. Oxford, Pergamon, 1964.

KENNEDY, J., EBERHART, R. C., Particle Swarm Optimization. *Proceedings of the 1995 IEEE International Conference on Neural Networks*, Perth, Australia, p. 1942-1948, 1995.

- 
- KIRKPATRICK, S., GELATT, C.D., VECCHI, M.P., Optimization by Simulated Annealing. *Science*, v.220, 4598, p.671-680, 1983.
- KLIR, G., FILGER, T., *Fuzzy Sets, Uncertainty, and Information*. Prentice-Hall, Englewood Cliffs, NJ, 1998.
- KRANTZ, S. G., PARKS, H. R., *The implicit function theorem : history, theory, and applications*. Birkhäuser, Boston, 2002.
- LEIDEMER, M. N., *Proposta de uma Metodologia de Otimização Evolucionária Robusta utilizando a Transformada Unscented Aplicável a Circuitos de RF/Microondas*. Dissertação (Mestrado), Universidade de Brasília, 2009.
- LEIDENFROST, R., ELMENREICH, W., Establishing Wireless Time-Triggered Communication using a Firefly Clock Synchronization Approach. *Proceedings of the 2008 International Workshop on Intelligent Solutions in Embedded Systems*, p. 1-18, 2008.
- LEWIS, A., Robust Regularization. *Mathematical Programming*. 2002.
- LIN, J. G. Multiple Objective Problems: Pareto-Optimal Solutions by Methods of Proper Equality Constraints. *IEEE Transactions on Automatic Control*, 21, n° 05, p. 641-650, 1976.
- LIMA, A.M.G. *Modelagem e otimização robusta de sistemas mecânicos em presença de amortecimento viscoelástico*. Tese (Doutorado). Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, 2007.
- LIMA, E. L. *Curso de Análise*. Volume 1. Edição n° 13, Projeto Euclides, Sociedade Brasileira de Matemática. Rio de Janeiro, 2011.
- LIMA, E. L. *Curso de Análise*. Volume 2. Edição n° 11, Projeto Euclides, Sociedade Brasileira de Matemática. Rio de Janeiro, 2012.
- LIMA, E. L. *Espaços Métricos*. Edição n° 5, Projeto Euclides, Sociedade Brasileira



de Matemática. Rio de Janeiro, 2013.

LIPSCHUTZ, S., LIPSON, M., *Álgebra Linear*. 3ª Edição. Coleção Schaum. São Paulo. Editora Brookman, 2004.

LIVIO, M., *The Golden Ratio: The Story of Phi, The World's Most Astonishing Number*. New York: Broadway Books, 2002.

LOBATO, F. S. *Otimização Multiobjetivo para o Projeto de Sistemas em Engenharia*. Tese (Doutorado). Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, 2008.

LOBATO, F. S., ARRUDA, E.B., BARROZO, M.A.S., STEFFEN, V.Jr. Estimation of drying parameters in rotary dryers using differential evolution. *Journal of Physics Conference Series*, 135, 2008.

LOBATO, F. S., STEFFEN Jr., V., Desenvolvimento de uma Estratégia de Otimização Auto-Adaptativa usando Modelos Caóticos de Busca e o Algoritmo Vagalume. *XIII Encontro de Modelagem Computacional*, Instituto Politécnico (IPRJ), Campus Regional da UERJ, Nova Friburgo/RJ, Brasil, 2010.

LOBATO, F. S., ARRUDA, E. B., CAVALINI Jr., A. Ap., STEFFEN Jr., V., Engineering System Design using Firefly Algorithm and Multi-objective Optimization. *Proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2011*, Washington, DC, USA, August 28-31, 2011a.

LOBATO, F. S., VIEIRA, L. G. M., BARROZO, M. A. S., Estudo do Desempenho de Hidrociclones usando a Metodologia de Superfície de Respostas e Otimização Multiobjetivo. *XIV Encontro de Modelagem Computacional, Instituto Politécnico (IPRJ)*, Campus Regional da UERJ, Nova Friburgo/RJ, Brasil, 2011b.

LOBATO, F. S., SANTOS, K. G., MALAGONI, R. A., Resolução de Problemas de

---

Controle Ótimo usando Técnicas de Otimização Bio-Inspiradas na Natureza. *XIX Congresso Brasileiro de Engenharia Química*, Búzios-RJ, 2012.

LUZ, E. F. P., BECCENERI, J. C., Campos Velho, H. F., Conceitualização do Algoritmo Vagalume e sua Aplicação na Estimativa de Condição Inicial da Equação de Calor. *IX Workshop do Curso de Computação Aplicada do INPE (WORCAP 2009)*, São José dos Campos, 2009.

MALISKA, C. R., *Transferência de Calor e Mecânica dos Fluidos Computacional*. Primeira edição. São Paulo - SP: LTC - Livros Técnicos e Científicos Editora S/A, 2004.

MARLER, R. T., ARORA, J. S. Survey of multi-objective optimization methods for engineering. *Structural Multidisciplinary Optimization*, v. 26, n. 1, p. 369-395, 2004.

MESSAC, A. Physical programming: Effective optimization for computational design. *AIAA Journal*, v. 34, n. 2, p. 149-158, 1996.

METROPOLIS, N., ULAM, S. *Journal of the American Statistical Association*. Vol. 44, nº 247, p. 335-341. 1949.

METROPOLIS, N. C., ROSENBLUTH, A. W., ROSENBLUTH, M N., TELLER, A. H., TELLER, E. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, v21, 6, p.1087-1092, 1953.

MICHALEWICZ, Z. *Genetic Algorithms + Data Structures = Evolution Programs*. 3ed. New York: Springer-Verlag, 1996.

MIETTINEN, K.M. *Nonlinear Multiobjective Optimization*. USA: Kluwer Academic Publishers, 1999.

MOREIRA, F.R., *O método de Newton. Uma Análise de Convergência Local e Semi-Local: O teorema de Kantorovich*. Dissertação (Mestrado). Instituto de Matemática e Estatística. Universidade Federal de Goiás, 2006.

MOREIRA, F. R., LOBATO, F. S., STEFFEN JR., V.; Otimização Multiobjetivo Robusta usando o Conceito de Média Efetiva e o Algoritmo de Colônia de Vagalumes. *XVI Encontro de Modelagem Computacional*, UESC, 2013a.

MOREIRA, F. R., LOBATO, F. S., STEFFEN JR., V.; Otimização Multiobjetivo Robusta com Restrições Usando o Algoritmo de Colônia de Vagalumes. *POSMEC - Simpósio do Programa de Pós-Graduação em Engenharia Mecânica*, UFU, 2013b.

MOSER, J., A new techniques for the construction of solutions of nonlinear differential equations. *Proc. Nat. Acad. Sci. USA*, 47, p. 1824-1831, 1961.

MYERS, R. H., *Response Surface Methodology*. First edition. Allyn and Bacon Inc.: Boston, 1971.

NASH, J., The embedding problem for Riemannian manifolds. *Ann. of Math*, 63 p.20-63, 1956.

NETO, A.J.S., BECCENERI, J.C., Técnicas de Inteligência Computacional Inspiradas na Natureza - Aplicação em Problemas Inversos em Transfênciã Radiativa. *Notas em Matemática Aplicada*, SBMAC, v41, 2009.

NULSSEALM, K.; TAVALAGE, J. Trade-off cut approach to multiple objective optimization. *Operations Research*, v. 6, p. 1424-1435, 1980.

OLIVEIRA, L.S. de *Uma Contribuição ao Estudo dos Métodos de Otimização Multi-objetivo*. Dissertação (Mestrado). Faculdade de Engenharia Mecânica. Universidade Federal de Uberlândia. 2005.

OLIVEIRA, G. T. S., *Estudo e Aplicações da Evolução Diferencial*. Dissertação (Mestrado), Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, 2006.

OLIVEIRA, G. T. S., *Projeto ótimo de robôs manipuladores 3r considerando a topologia do espaço de trabalho*. Dissertação (Mestrado), Faculdade de Engenharia

- Mecânica, Universidade Federal de Uberlândia, 2012.
- OPPENHEINER, K. R. A proxy approach to multi-attribute decision making. *Management Science*, v. 15, p. 309-333, 1978.
- ORTEGA, J.M., RHEINBOLDT, W.G. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York, 1970.
- OSYCZKA, A. An approach to multicriterion optimization problems for engineering design. *Comp. Methods in Applied Mech. and Eng.*, v. 15, p. 309-333, 1978.
- OSYCZKA, A. *Multicriterion Optimization in Engineering with Fortram Programs*. First Edition. England: Ellis Horwood Limited, 1984.
- OSYCZKA, A. Multicriterion Decision Making with Min-Max Approach. *Optimization Methods in Structural Design*, Edited by H. Eschenauer, N. Olhoff, Euromech-Colloquium 164, Wissenschaftsverlag, 1989.
- OSYCZKA, A. Computer Aided Multicriterion Optimization System (CAMOS): Software Package in FORTRAN. *International Software Publishers*, 1992.
- PAENK, I., BRANKE, J., JIN, Y., Efficient Search for Robust Solutions by Means of Evolutionary Algorithms and Fitness Approximation. *IEEE Transactions on Evolutionary Computation*. Vol. 10, n° 4, 2006.
- PALLI, N., AZRAM, S., MCCLUSKEY, P., SUNDARARAJAN, R. An interactive multistage  $\epsilon$ -inequality constraint method for multiple objectives decision making. *ASME Journal of Mechanical Design*, v. 120, n. 4, p. 678-686, 1999.
- PARETO, V. *Cours D'Economie Politique*. First Edition. França. Vol. I and II. F. Rouge, Lausanne, 1896.
- PARMEE, I.C., Cluster-oriented genetic algorithms for the identification of high-performance regions of design spaces. In: *Proceedings of EvCA96*, 1996.
- PRICE, K. V., STORN, R.M., LAMPINEN, J. A., *Differential Evolution - A Practical*

---

*Approach to Global Optimization*. Springer - Natural Computing Series, 2005.

PURCINA, L.A., *Técnicas de Otimização Evolutiva Aplicadas a Solução de Grandes Sistemas Lineares*. Tese (Doutorado). Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, 2010.

QUITZRAU, L. E., *Análise Dinâmica de Rotores de Grupos Turbo e Hidrogeradores com o Método das Matrizes de Transferência*. Dissertação (Mestrado). Escola de Engenharia. Universidade Federal do Rio Grande do Sul. 2002.

RAY, T., TSAI, H. M., A parallel Hybrid optimization algorithm for robust airfoil design. *AIAA Aerospace Sciences Meeting and Exhibit*. p. 11474-11482, 2004.

RAYWARD-SMITH, V. J., OSMAN, I. H., REEVES, C.R., SMITH, G.D., *Modern Heuristic Search Methods*. John Wiley and Sons Ltd, Baffins Lane, Chichester, West Sussex, England, 1996.

REEVES, C. *Genetic Algorithms, Handbook of Metaheuristics*. Glover, F. e Kochenberger, G.A.(Ed.), International Series in Operations Research and Management Science, v.57. New York: Springer, 2003.

ROJAS ,J.E., VIANA, F.A.C., RADE, D.A., STEFFEN Jr., V. Identification of External Forces in Mechanical Systems by Using LifeCycle Model and Stress-Stiffening Effect. *Mechanical Systems and Signal Processing*, Vol. 21 (7), p. 2900-2917, 2007.

SADEK, I.S., BRUCH, J. C. Jr., SLOSS, J.M., ADALI, S. Structural Control of the Variable Cross-Section Beam by Distributed Forces. *Mechanics of Structures and Machines 16*, Nº 03, p. 313-333, 1989.

SAKAWA, M. An Interactive Computer Program for Multiobjective Decision Making by Proxy Optimization Technique. *Int. J. Man-Machine Studies*, 14, p. 193-213, 1981.

SALDARRIAGA, V. M., STEFFEN Jr., V., DER HAGOPIAN, J., MAHFOUD,

- J., On the Balancing of Flexible Rotating Machines by Using Inverse Problem Approach. *Journal of Vibration and Control*, v.17, p.1021-1033, 2011.
- SALUKVADZE, M. E. On existence of solution in problems of optimization under vector valued criteria. *Journal of Optimization Theory and Applications*, v. 12, p. 203-217, 1974.
- SAMPAIO, P.R. *Teoria, Métodos e Aplicações de Otimização Multiobjetivo*. Dissertação (Mestrado). Instituto de Matemática e Estatística, Universidade de São Paulo, 2011.
- SANTOS, M. I. C. S. R., *Construção de Metamodelos de Regressão Não Linear para Simulação de Acontecimentos Discretos*. Tese (Doutorado). Instituto Superior Técnico - Universidade Técnica de Lisboa, 2002.
- SARAMAGO, S. de F. P. *Otimização da Trajetória de Robôs Manipuladores na Presença de Obstáculos*. Tese (Doutorado). Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, 1998.
- SAWARAGI, Y., NAKAYAMA, H., TANINO, T. *Teory of the Multiobjective Optimization*. Academic Press, Inc., Orlando, Flórida, 1985.
- SCHAFFER, J.D. *Multi Objective Optimization with Vector Evaluated Genetic Algorithms*. Tese (Doutorado). Vanderbilt University, 1984.
- SERAPIÃO, A. B. de S., Fundamentos de Otimização por Inteligência de Enxames: Uma Vizão Geral. *Revista Controle & Automação*. Vol. 20, nº 3, p. 271-304, 2009.
- SIGLER, L.E. *Fibonacci's Liber Abaci: A Translation into Modern English of Leonardo Pisano's Book of Calculation (Sources and Studies in the History of Mathematics and Physical Sciences)*. Springer-Verlag. 2003.
- SILVA, V.V. da. Ordenamento do Corpo dos Complexos  $\mathbb{C}$ . *Revista da Olimpíada de Matemática do Estado de Goiás* (OMEG). IME-UFG. Editora da UFG. n.3, p.

106-109, abr. 2002.

SILVA, C.M. *Desenvolvimento de um Algoritmo de Otimização Multiobjetivo usando Algoritmos Genéticos*. Tese (Doutorado). Universidade Federal do Rio de Janeiro, 2004. SOUZA, D. L., GEDRAITE, R., LOBATO, F. S., A Comparative Study using Bio-Inspired Optimization Methods Applied to Controllers Tuning. *Frontiers in Advanced Control Systems*, Intech, p.7-27, 2012.

SRINIVAS, N., DEB, K. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*. v.2, n.3, p.221-248, 1994.

STADLER, W. Caveats and boons of multicriteria optimization. *Microcomput. Civ. Eng.*, v. 10, n. 1, p. 291-299, 1995.

STEFFEN Jr., V., ASSIS, E. G., LEPORE NETO, F. P., Multicriterion Techniques for the optimization of rotors.In: *Multicriterion Techniques for the Optimization of Rotors*, v.2, p.236-249, 1999.

STORN, R., PRICE, K., Differential Evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces.*Technical Report TR-95-012, International Computer Science Institute*, Berkeley. 1995.

STORN, R., PRICE, K., Differential Evolution: a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11, p.341-359, 1997.

TESCHKE,G., BORRIES, C. Accelerated projected steepest descent method for nonlinear inverse problems with sparsity constraints.*Inverse Problems*, Vol. 20, Number 2, 2010.

THOMPSON, A., Evolutionary Techniques for fault tolerance.In: *Proc. UKACC International Conference Control*. p. 693-698, 1996.

TSUZUKI, M. de S. G., *Simulated Annealing - Single and Multiple Objective Problems*.

Intech, 2012.

VANDERPLAATS, G. N. *Numerical Optimization Techniques for Engineering Design*. Third Edition. USA: VR D INC. Colorado Springs, 1999.

VENTER, G., SOBIESZCZANSKI-SOBIESKI, J., Particle Swarm Optimization. *Proceedings of the 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Denver, CO, Vol. AIAA - 2002-1235, April 22-25, 2002.

VIANA, F. A. C. *Surrogate Modeling Techniques and Optimization Methods Applied to Design and Identification Problems*. Tese(Doutorado), Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, 2008.

VIANA, F.A.C., VENTER, G., BALABANOV, V. An Algorithm for Fast Optimal Latin Hypercube Design of Experiments. *International Journal for Numerical Methods in Engineering*. Vol. 82, Issue 2, p. 135-156. 2010.

VICENTE, B. G. G. L. Z., *Projeto Robusto de Circuitos Shunt para o Controle Passivo de Vibrações de Estruturas Compostas*. Dissertação (Mestrado). Faculdade de Engenharia Mecânica. Universidade Federal de Uberlândia, 2014.

WALZ, F. M. An engineering approach: Hierarchical optimization criteria. *IEEE Trans Automatic Control*, v. 12, p. 179-191, 1967.

WANG, F. S. SU, T. L. JANG, H.J. Hybrid differential evolution for problems of kinetic parameter estimation and dynamic optimization of an ethanol fermentation. *Proceedingsess, Industry Engineering Chemical Research*, 40, p.2876-2885, 2001.

WENDELL,R.E., LEE, D.N. Efficiency in Multiple Objective Optimization Problems. *Mathematical Programming*, 12, nº 3, p. 406-414, 1977.

WERNER-ALLEN, G., TEWARI, G., PATEL, A., WELSH, M., and NAGPAL, R., Firefly-Inspired Sensor Network Synchronicity with Realistic Radio Effects. *Proceedings of the Third International Conference on Embedded Networked Sensor*



---

*Systems*, p.142-153, 2005.

WIESMANN, D., HAMMEL, U., BÄCK, T., Robust Design Multilayer optical coatings by means of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 2(4), p. 162-167, 1998.

YAMAGUCHI, Y., ARIMA, T., Aerodynamic optimization for the transonic compressor stator blade. *Optimization in Industry*, I.C. Parmee and P. Hajela, Eds.: Springer, p. 163-182, 2002.

YANG, X.-S. *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, Cambridge, 2008.

YPMA, T, J., Historical Developement of the Newton-Raphson Method. *SIAM Review*, 37(4), p.531-551, 1995.

YU, P.L. A Class of Solutions for Group Decision Problems. *Management Science*,19, n° 8, p.936-946, 1973.

ZELENY, M. *Compromise Programming, Multiple Criteria Decision Making*. Edited by J.L. Cochrane, M. Zeleny, University of South Carolina Press, Columbia, South Carolina, p.262-301, 1973.

ZITZLER, E. *Evolutionary algorithms for multiobjective optimization: methods and applications*. Tese (Doutorado), Federal Institute of Technology Zurich, Zurich, Swiss, 1999.

ZITZLER, E., DEB, K., THIELE, L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation Journal*, v.8, n.2, p.125-148, 2000.