

Ivan Reinaldo Meneghini

Uma Proposta de Algoritmo Baseado em  
Cone de Preferência para Otimização com  
Muitos Objetivos e Robusta

Brasil

07/12/2018



Ivan Reinaldo Meneghini

# Uma Proposta de Algoritmo Baseado em Cone de Preferência para Otimização com Muitos Objetivos e Robusta

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do Título de Doutor em Engenharia Elétrica.

Universidade Federal de Minas Gerais

Programa de Pós-Graduação em Engenharia Elétrica - PPGEE

Machine Intelligence and Data Science Laboratory – MINDS

Orientador: Frederico Gadelha Guimarães (UFMG)

Coorientador: Antonio Gaspar-Cunha (U. Minho)

Brasil

07/12/2018

Ivan Reinaldo Meneghini

Uma Proposta de Algoritmo Baseado em Cone de Preferência para Otimização com Muitos Objetivos e Robusta/ Ivan Reinaldo Meneghini. – Brasil, 07/12/2018-186 p. : il. (algumas color.) ; 30 cm.

Orientador: Frederico Gadelha Guimarães (UFMG)

Tese (Doutorado) – Universidade Federal de Minas Gerais

Programa de Pós-Graduação em Engenharia Elétrica - PPGEE

Machine Intelligence and Data Science Laboratory – MINDS , 07/12/2018.

1. Otimização multi-objetivo Robusta. 2. ROI. 3. Visualização de dados. 4. Problemas de Otimização Multiobjetivo. 5. Geração de vetores de Peso

I. Frederico Gadelha Guimarães. II. Universidade Federal de Minas Gerais. III. Escola de Engenharia. IV. Uma Proposta de Algoritmo Baseado em Cone de Preferência para Otimização com Muitos Objetivos e Robusta

*Dedico este trabalho à memória dos meus pais,  
Herculano Nelo Meneghini e Vera Reynaldo Meneghini,  
e à minha filha,  
Júlia Javoski Meneghini.*



# Agradecimentos

Inicialmente gostaria de agradecer ao professor e amigo Frederico cuja presença e valorosos conselhos foram de extrema importância e ao professor Gaspar que me acolheu em Portugal em um momento delicado de minha vida e fez inestimáveis contribuições a este trabalho.

Tenho uma enorme dívida com meus amigos e familiares. Senti muito a ausência de todos. Tem uma parte de vocês naquilo que sou hoje.

A todos os membros do MINDS, o melhor laboratório do mundo!

Ao Instituto Federal de Minas Gerais – IFMG, em particular aos amigos do *campus* de Formiga, cujo apoio e suporte recebido foi fundamental para o desenvolvimento deste trabalho.

À Universidade Federal de Minas Gerais – UFMG, em particular a todos do Programa de Pós-graduação em Engenharia Elétrica, cuja excelência é maior do que qualquer palavra.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – CAPES, cujo suporte financeiro tornou este trabalho possível.



*Mar calmo não faz bom marinheiro.*



# Resumo

A solução de um problema de otimização multiobjetivo é um conjunto caracterizado pelo *trade off* entre  $M$  objetivos. No caso do problema de minimização  $F : \mathbb{R}^N \rightarrow \mathbb{R}^M$ , estas soluções de compromisso correspondem a um conjunto minimal segundo uma relação de ordem parcial no espaço  $\mathbb{R}^M$ . A busca por soluções de menor variação na presença de ruído nas variáveis  $\mathbf{x} \in \mathbb{R}^N$  caracteriza a Otimização Multiobjetivo Robusta. Este trabalho apresenta um algoritmo coevolutivo para Otimização Robusta. A proposta utiliza a estratégia de decomposição/agregação do espaço dos objetivos em um algoritmo coevolutivo competitivo. Em conjunto com esta nova técnica foi desenvolvido um novo método de geração de vetores de peso quase igualmente espaçados no espaço dos objetivos. Este novo método de geração de vetores de peso não apresenta limitação quanto ao número de vetores de peso criados nem quanto à norma de cada vetor, que podem estar localizados no primeiro ortante do espaço dos objetivos ou formar um cone de vetores com vértice na origem. O eixo deste cone corresponde a um vetor de preferências do tomador de decisão e sua abertura define a extensão da região de interesse escolhida. A qualidade dos vetores de peso produzidos por esta nova metodologia foi comparada com o método usual de geração de vetores de peso e os resultados se mostraram satisfatórios. Adicionalmente uma nova classe de problemas de otimização multiobjetivo foi desenvolvida, abrangendo otimização contínua e robusta, com e sem a presença de restrições de igualdade e desigualdade. Seguindo a estrutura utilizada na construção das funções de teste, uma nova métrica de avaliação de desempenho também é apresentada. A comparação dos resultados obtidos entre o método proposto e outras técnicas mostrou a superioridade dos métodos apresentados. Uma amostra dos resultados obtidos foi utilizada na ferramenta de visualização de dados desenvolvida, ilustrando as conclusões obtidas.

**Palavras-chave:** Otimização Multiobjetivo Robusta. Região de Interesse. Problemas de Otimização Multiobjetivo Robusta. Visualização de Dados.



# Abstract

The solution of a multi-objective optimization problem is a set characterized by the trade off of  $M$  objectives. In the case of the minimization problem  $F : \mathbb{R}^N \rightarrow \mathbb{R}^M$ , these trade off solutions correspond to a minimal set according to a partial order relation in space  $\mathbb{R}^M$ . The search for solutions of smaller variation in the presence of noise in the variables  $\mathbf{x} \in \mathbb{R}^N$  characterizes Robust Multiobjective Optimization. This work presents a co-evolutionary algorithm for Robust Optimization. The presented proposal uses the objective space decomposition/aggregation strategy in a competitive co-evolution algorithm. Along with this new technique, a new method of generating vectors of weight almost equally spaced in the objective space was developed. This new method of generating weight vectors is not limited in the number of weight vectors created neither to the norm of each vector, that can be located in the first orthant of the objective space to form a cone of vectors with vertex in the origin. The axis of this cone corresponds to a preference vector of the decision maker and its aperture defines the extension of the chosen region of interest. The quality of the weight vectors of weight produced by this new methodology was compared with the usual method of generation of weight vectors and the results were satisfactory. In addition, a new class of multi-objective optimization problems was developed, encompassing usual and robust optimization, with and without the presence of equality and inequality constraints. Following the structure used in the construction of the test functions, a new performance evaluation metric is also presented. The comparison of the results obtained between the proposed method and other techniques showed the superiority of the presented methods. A sample of the results obtained was used in the data visualization tool developed, showing the conclusions obtained.

**Keywords:** Robust Multiobjective Optimization. Region of Interest. Robust Multiobjective Optimization Problem. Data Visualization



# Lista de ilustrações

Figura 1 – Etapas do processo de tomada de decisão. . . . .	29
Figura 2 – Uma função com máximos e mínimos locais . . . . .	31
Figura 3 – Elementos da otimização multiobjetivo . . . . .	40
Figura 4 – Recombinação binária de um ponto . . . . .	43
Figura 5 – <i>Penalty Boundary Intersection</i> (PBI) . . . . .	48
Figura 6 – <i>Non Dominating Sorting</i> . . . . .	51
Figura 7 – Soluções do sistema de equações (3.1) com as condições iniciais $a$ e $b$ . . . . .	56
Figura 8 – Diagrama de parâmetros . . . . .	57
Figura 9 – Solução robusta e solução global . . . . .	61
Figura 10 – Medida de robustez pelo valor médio da função. . . . .	64
Figura 11 – Media de robustez pela variância . . . . .	65
Figura 12 – Medida de robustez pelo pior caso. . . . .	65
Figura 13 – Amostragem por cubo latino ortogonal . . . . .	68
Figura 14 – Fronteira Pareto robusta e usual . . . . .	70
Figura 15 – Variação da medida de robustez . . . . .	72
Figura 16 – Função $g(\mathbf{x})$ utilizada nos problemas DTLZ1 a DTLZ4 restrita a uma variável $0 \leq x \leq 1$ . . . . .	80
Figura 17 – Diferentes valores do parâmetro $t$ da meta-variável $y$ aplicada no pro- blema DTLZ2 . . . . .	83
Figura 18 – Pontos com diferentes valores de $p$ -norma no espaço $\mathbb{R}^2$ . . . . .	85
Figura 19 – Função deceptiva $g(\mathbf{x})$ em uma variável com $r = 0.1, 0.05$ e $0.01$ . . . . .	88
Figura 20 – DTLZ2 com 2 objetivos utilizando a função deceptiva $g(\mathbf{x})$ e $r =$ $0.1, 0.05$ e $0.01$ . . . . .	88
Figura 21 – A função $g(\mathbf{x})$ robusta restrita a uma variável $0 \leq x \leq 1$ . . . . .	89
Figura 22 – Efeito da presença de ruído em uma vizinhança dos valores ótimos globais e robustos para a função $g(\mathbf{x})$ robusta. . . . .	89
Figura 23 – Fronteira Pareto do problema pd-MOP-01 não dissimilar com 2 e 3 objetivos. . . . .	92
Figura 24 – Valor de $\frac{1}{\sqrt[p]{M}}$ em espaços $M$ -dimensionais . . . . .	93
Figura 25 – Fronteira Pareto do problema pd-MOP-02 não dissimilar com 2 e 3 objetivos com $p = 3$ . . . . .	93
Figura 26 – Fronteira Pareto do problema pd-MOP-03 não dissimilar com 2 e 3 objetivos. . . . .	94
Figura 27 – Carte figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812-1813 . . . . .	97
Figura 28 – Duas representações para o mesmo conjunto de pontos . . . . .	98

Figura 29 – Representação de um mesmo ponto no sistema de Coordenadas Estelares e no método RadViz. Figura retirada de Rubio-Sanchez et al. [125]. . . . .	99
Figura 30 – Diagrama de cordas de um conjunto de pontos no espaço $\mathbb{R}^{10}$ . Os objetivos estão indexados de 0 a 9 . . . . .	101
Figura 31 – Três conjuntos de pontos representados utilizando o gráfico de dispersão de pontos e o mapeamento angular em espaços com duas e três dimensões	102
Figura 32 – O mesmo conjunto de pontos representados no diagrama de dispersão de pontos no CAP-Vis e sua classificação pela aproximação angular com os vetores da base canônica e seus elementos gráficos: A: Diagrama de cordas; B: Coordenadas Paralelas; C: Mapeamento angular; D: Índice da região representada. . . . .	104
Figura 33 – Pontos em superfícies de norma (quasi norma) constante no espaço $\mathbb{R}^3$	104
Figura 34 – Soluções obtidas no problema dissimilar pd-MOP-02 com 5 objetivos .	106
Figura 35 – Histograma das soluções obtidas no problema dissimilar pd-MOP-02 com 5 objetivos . . . . .	107
Figura 36 – CAP-vis aplicado ao problema robusto pd-MOP-02 com 7 objetivos . .	109
Figura 37 – Dois conjuntos de vetores de peso no espaço $\mathbb{R}^3$ gerados pelo método proposto. A linha vermelha representa a interseção das superfícies de $p$ -norma constante com o bordo do primeiro ortante. . . . .	115
Figura 38 – Os elementos de um cone e dois conjuntos de cones de vetores nos espaços $\mathbb{R}^2$ e $\mathbb{R}^3$ . . . . .	117
Figura 39 – Boxplot da evolução do valor médio da distância entre um vetor de peso $\mathbf{w}_i$ e seu vizinho mais próximo em espaços de dimensão 5,10 e 20. . . . .	119
Figura 40 – Dendrograma do processo de formação de agrupamentos de vetores nos espaços $\mathbb{R}^5$ , $\mathbb{R}^{10}$ e $\mathbb{R}^{20}$ . . . . .	120
Figura 41 – Distância de fusão entre agrupamentos de vetores nos espaços $\mathbb{R}^5$ , $\mathbb{R}^{10}$ e $\mathbb{R}^{20}$ . A escala horizontal indica o ordinal do conjunto de vetores de peso.	121
Figura 42 – Dois conjuntos de vetores de peso no espaço $\mathbb{R}^{10}$ gerados por diferentes métodos. . . . .	122
Figura 43 – Dois cones de vetores de peso no espaço $\mathbb{R}^{10}$ com diferentes eixos. . . . .	123
Figura 44 – Soluções obtidas pelo algoritmo MOEA/D e NSGAIII para o problema pdMOP02 com valor de $p$ -norma igual a 2 e 3 objetivos. . . . .	127
Figura 45 – Soluções obtidas pelo algoritmo MOEA/D e NSGAIII para o problema pdMOP02 com 5 objetivos . . . . .	128
Figura 46 – Soluções obtidas pelo algoritmo MOEA/D para o problema pdMOP02 com 10 objetivos . . . . .	129
Figura 47 – Coordenadas paralelas das soluções obtidas pelo algoritmo NSGAIII para o problema pdMOP02 com 5 objetivos . . . . .	130

Figura 48 – Coordenadas paralelas das soluções obtidas pelo algoritmo MOEA/D para o problema pdMOP02 com 5 objetivos . . . . .	130
Figura 49 – Distribuição valor médio do número de soluções encontradas na ROI em todos os cenários . . . . .	142
Figura 50 – Soluções obtidas pelos algoritmos coneMOEA/D e coneNSGA-III no problema pd-MOP02 com 3 objetivos. . . . .	144
Figura 51 – Soluções obtidas pelos algoritmos coneMOEA/D e coneNSGA-III no problema pd-MOP02 com 5 objetivos. . . . .	145
Figura 52 – Soluções obtidas pelos algoritmos coneMOEA/D e coneNSGA-III no problema pd-MOP02 com 10 objetivos. . . . .	146
Figura 53 – Ação das duas instâncias de otimização E1 e E2 em um vetor $A$ no espaço dos objetivos. . . . .	152
Figura 54 – Soluções obtidas pelo algoritmo NSGA-III no problema robusto pdMOP02.	158
Figura 55 – Soluções obtidas pelo algoritmo CR-MOEA/D no problema robusto pdMOP02. . . . .	158
Figura 56 – Soluções obtidas pelo algoritmo NSGA-III na ROI R1 do problema robusto pdMOP02. . . . .	159
Figura 57 – Soluções obtidas pelo algoritmo CR-MOEA/D na ROI R1 do problema robusto pdMOP02. . . . .	159
Figura 58 – Soluções obtidas pelo algoritmo NSGA-III na ROI R2 do problema robusto pdMOP02. . . . .	161
Figura 59 – Soluções obtidas pelo algoritmo CR-MOEA/D na ROI R2 do problema robusto pdMOP02. . . . .	161
Figura 60 – Soluções obtidas pelo algoritmo NSGA-III na ROI R3 do problema robusto pdMOP02. . . . .	162
Figura 61 – Soluções obtidas pelo algoritmo CR-MOEA/D na ROI R3 do problema robusto pdMOP02. . . . .	162
Figura 62 – Elementos do motor de ímã permanente . . . . .	164
Figura 63 – Fronteira Pareto e solução robusta do problema do motor de ímã permanente . . . . .	165



# Lista de tabelas

Tabela 1	– Parâmetros dos vetores de peso gerados nos experimentos realizados . . .	117
Tabela 2	– Média, desvio padrão e coeficiente de variação da distância entre um vetor de peso $\mathbf{w}_i$ e seu vizinho mais próximo em espaços de dimensão 5,10 e 20. . . . .	118
Tabela 3	– Parâmetros utilizados na geração de vetores pelo método de Scheffé . . .	120
Tabela 4	– Média, desvio padrão e coeficiente de variação de Pearson da distância entre um vetor de peso $\mathbf{w}_i$ e seu vizinho mais próximo. . . . .	122
Tabela 5	– Valores médios e variância das métricas Hiper-Volume (HV), <i>Inverted Generational Distance</i> (IGD) e a Métrica de Distância Radial (RDM) obtidas em 30 instâncias dos algoritmos MOEA/D e NSGAIII utilizando vetores de peso gerados pelo método de Scheffé (MOEA/D e NSGAIII) e pelo método proposto (wv-MOEA/D e wv-NSGAIII) . . . . .	125
Tabela 6	– p-valores para o teste de Wilcoxon pareado para as métricas utilizadas nos algoritmos MOEA/D e NSGAIII com as diferentes metodologias de geração de vetores de peso . . . . .	126
Tabela 7	– p-valores para o teste de Wilcoxon pareado para as métricas utilizadas nos algoritmos MOEA/D e NSGAIII com as diferentes metodologias de geração de vetores de peso . . . . .	126
Tabela 8	– Características dos eixos $\mathbf{v}_i$ e ângulos de abertura $\theta_j$ dos cones de preferências utilizados nos experimentos . . . . .	136
Tabela 9	– Valores médios das métricas IGD e RDM obtidas nos experimentos para os cones definidos pelo ângulo de abertura $\theta_1$ . . . . .	139
Tabela 10	– Valores médios das métricas IGD e RDM obtidas nos experimentos para os cones definidos pelo ângulo de abertura $\theta_2$ . . . . .	140
Tabela 11	– Valores médios das métricas IGD e RDM obtidas nos experimentos para os cones definidos pelo ângulo de abertura $\theta_3$ . . . . .	141
Tabela 12	– Regiões de Interesse utilizadas nos testes realizados . . . . .	153
Tabela 13	– Valores dos parâmetros utilizados nos experimentos . . . . .	154
Tabela 14	– Métricas IGD e RDM para o problema robusto poMOP02 com e sem a presença de ruído na variável $\mathbf{x}^d$ . . . . .	155
Tabela 15	– Tamanho da população encontrada na ROI e nas vizinhanças R1 e R2 obtida pelos algoritmos CR-MOEA/D no problema robusto pdMOP02 . . . . .	156
Tabela 16	– Métricas IGD e RDM para as soluções obtidas pelo algoritmo CR-MOEA/D no problema robusto pdMOP02 com e sem a presença de ruído na variável $\mathbf{x}^d$ . . . . .	157
Tabela 17	– Parâmetros de simulação do motor de ímã permanente . . . . .	163

Tabela 18 – Descrição das variáveis do problema de Otimização do motor de ímã permanente. . . . .	164
Tabela 19 – Objetivos do problema de otimização do motor de ímã permanente. . .	165

# Lista de Algoritmos

1	Algoritmo Genético básico . . . . .	41
2	MOEA/D . . . . .	50
3	NSGA-III . . . . .	53
4	Geração de vetores de peso . . . . .	114

\*



## Acrônimos

**DM** Decision Maker

**MOEA** Multi-Objective Evolutionary Algorithm

**ROI** Region Of Interest

**ASF** Achievement Scalarizing Functions

**RVEA** *Reference Vector Guided Evolutionary Algorithm*

**AE** Algoritmo Evolutivo

**AG** Algoritmo Genético

**AC** Algoritmo Coevolucionário

**IPM** *Interior Permanent Magnet*

**MOEA** *Multi-Objective Evolutionary Algorithm*

**NSGA** Non-Dominated Sorting Genetic Algorithm

**PBI** *Penalty Boundary Intersection*

**POM** Problema de Otimização Multiobjetivo

**PPGEE** Programa de Pós Graduação em Engenharia Elétrica

**SBX** Simulated Binary Crossover

**TQM** *Total Quality Management*

**HV** *Hyper-Volume Indicator*

**IGD** *Inverted Generational Distance*

**RDM** *Radial Distance Metric*

**PF** Pareto Front

**RPF** Robust Pareto Front



# Lista de símbolos

$\mathbb{R}^M$ :	Espaço dos Objetivos
$M$ :	Número de Objetivos
$\mathbb{R}^N$ :	Espaço de decisão
$N$ :	Número de variáveis de decisão
$\mathbf{x}$ :	Vetor do espaço de decisão
$\Delta$ :	Espaço de vetores de ruído
$\delta$ :	Vetor de ruído
$\delta_i$ :	$i$ -ésima coordenada do vetor de ruído
$x_i$ :	$i$ -ésima componente do vetor $\mathbf{x}$
$F(\mathbf{x})$ :	Vetor no espaço dos objetivos
$f_i(\mathbf{x})$ :	$i$ -ésimo objetivo do vetor $F(\mathbf{x})$
$g_i(\mathbf{x})$ :	Restrição de desigualdade
$h_i(\mathbf{x})$ :	Restrição de igualdade
$\mathcal{S}$ :	Conjunto dos pontos factíveis do espaço de decisão
$\preceq$ :	Relação de dominância segundo Pareto
$\mu_c$ :	Fator de dispersão do operador de recombinação SBX
$\mu_m$ :	Fator de dispersão do operador de mutação polinomial
$W$ :	Conjunto de vetores de peso
$\mathbf{w}$ :	Vetor de peso
$w_i$ :	$i$ -ésima componente do vetor de peso $\mathbf{w}$
$\mathbf{z}$ :	Ponto utópico do espaço de decisão
$\ \mathbf{v}\ _p$ :	$p$ - norma (ou norma $\ell_p$ ) do vetor $\mathbf{v}$
$B(i)$ :	Conjunto de vetores $\mathbf{x}_j$ associados ao vetor $\mathbf{x}_i$

$R(\mathbf{x}, \delta)$ :	Medida de Robustez da variável $\mathbf{x}$ com parâmetro $\delta$
$\mathbf{x}_p$ :	Vetor responsável pelo posicionamento dos objetivos no espaço $\mathbb{R}^M$
$\mathbf{x}_d$ :	Vetor responsável pela convergência dos objetivos no espaço $\mathbb{R}^M$
$F_p(\mathbf{x})$ :	Função responsável pelo posicionamento dos objetivos no espaço $\mathbb{R}^M$
$F_d(\mathbf{x})$ :	Função responsável pela convergência dos objetivos no espaço $\mathbb{R}^M$
$\mathbf{v}$ :	Eixo do cone de vetores

# Sumário

1	INTRODUÇÃO	29
1.1	Caracterização do problema	29
1.2	Objetivos	32
1.3	Contribuições da tese	33
1.4	Organização do trabalho	35
2	OTIMIZAÇÃO	37
2.1	Introdução	37
2.2	Algoritmos Evolutivos	40
2.2.1	Técnicas avançadas em Computação evolutiva	45
2.2.1.1	Nicho	45
2.2.1.2	Autoadaptação	45
2.2.1.3	Coevolução	46
2.2.1.4	Decomposição/Agregação	46
2.2.2	MOEA/D	47
2.2.3	NSGA-III	50
2.3	Considerações finais	52
3	OTIMIZAÇÃO ROBUSTA	55
3.1	Incertezas no processo de Otimização	58
3.2	Medidas de robustez	62
3.3	Incorporação da robustez no processo de otimização	65
3.4	Otimização Robusta Multiobjetivo	69
3.5	Considerações finais	71
4	UM NOVO CONJUNTO DE FUNÇÕES DE TESTE	73
4.1	ZDT	74
4.2	DTLZ	74
4.3	WFG	78
4.4	Limitações das funções	79
4.5	Uma nova família de funções de teste	81
4.5.1	A função $F_p(\mathbf{x})$	82
4.5.2	A função $F_d(\mathbf{x})$	86
4.5.2.1	A função $\phi(\mathbf{x})$	86
4.5.2.2	A função $g(\mathbf{x})$	87
4.5.3	Restrições de igualdade e desigualdade	90

4.5.4	Funções propostas . . . . .	90
4.5.4.1	pd-MOP-01 . . . . .	91
4.5.4.2	pd-MOP-02 . . . . .	92
4.5.4.3	pd-MOP-03 . . . . .	94
4.6	Métrica de Distância Radial . . . . .	94
4.7	Considerações finais . . . . .	95
5	<b>VISUALIZAÇÃO DE DADOS EM ESPAÇOS DE DIMENSÃO ELE-</b> <b>VADA . . . . .</b>	<b>97</b>
5.1	CAP-vis . . . . .	103
5.2	Aplicações . . . . .	105
5.2.1	Visualização de soluções . . . . .	105
5.2.2	Comparação de soluções . . . . .	106
5.2.3	Considerações finais . . . . .	108
6	<b>GERAÇÃO DE VETORES DE PESO EM ALGORITMOS EVO-</b> <b>LUTIVOS . . . . .</b>	<b>111</b>
6.1	Introdução . . . . .	111
6.2	Método proposto . . . . .	112
6.2.1	Cone de vetores . . . . .	116
6.3	Análise do método proposto . . . . .	116
6.3.1	Experimento 1 . . . . .	116
6.3.2	Experimento 2 . . . . .	124
6.4	Considerações finais . . . . .	127
7	<b>INCORPORAÇÃO DE REGIÃO DE INTERESSE EM PROBLE-</b> <b>MAS DE OTIMIZAÇÃO MULTIOBJETIVO . . . . .</b>	<b>133</b>
7.1	Introdução . . . . .	133
7.2	Método Proposto . . . . .	135
7.3	Considerações finais . . . . .	146
8	<b>UM ALGORITMO COEVOLUTIVO PARA OTIMIZAÇÃO RO-</b> <b>BUSTA MULTIOBJETIVO . . . . .</b>	<b>149</b>
8.1	Resultados . . . . .	153
8.2	Otimização Robusta do motor IPM . . . . .	163
8.3	Considerações Finais . . . . .	166
9	<b>CONCLUSÃO . . . . .</b>	<b>167</b>
9.1	Trabalhos futuros . . . . .	168

Referências . . . . .	171
-----------------------	-----



# 1 Introdução

## 1.1 Caracterização do problema

O dicionário Priberam em sua versão digital <sup>1</sup> define otimizar como “dar a uma máquina, a uma empresa, a uma ação, etc., o rendimento ótimo, criando as condições mais favoráveis ou tirando o melhor partido possível”. Uma das etapas dos processos de gestão e melhoria da qualidade consiste na criação de modelos matemáticos que possibilitem a implementação de processos de otimização que auxiliem o tomador de decisão (Decision Maker (DM)) a escolher a melhor estratégia. A figura 1 ilustra o procedimento geral em um processo de tomada de decisão, que consiste nas seguintes etapas [140]:

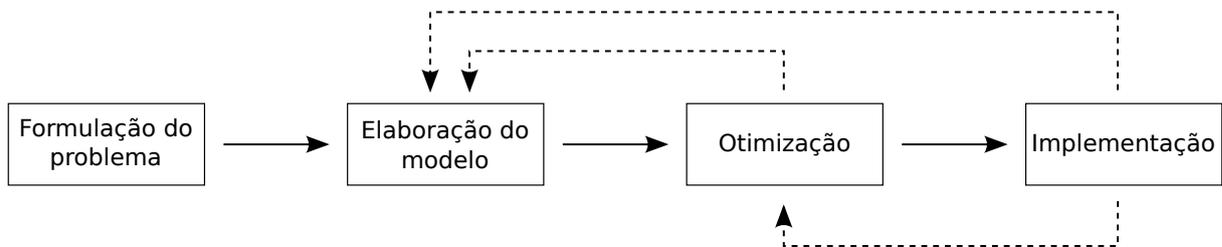


Figura 1 – Etapas do processo de tomada de decisão.

Observe que o fluxograma descrito na figura 1 não é unidirecional: ajustes no modelo elaborado podem ser feitos após as etapas de otimização ou de implementação das soluções escolhidas, que também pode ser utilizado para aprimorar o processo de otimização. As etapas do processo de tomada de decisão podem ser descritas da seguinte maneira:

### A. Formulação do Problema:

Nesta etapa deve-se definir com clareza e precisão o problema e suas variáveis, identificando fatores que possam provocar algum tipo de ruído ou variação em todas as etapas envolvidas. Nesta etapa também se deve definir que tipo de resposta se deseja. Esta etapa inicial é de extrema importância uma vez que dela dependem todas as demais.

### B. Elaboração do Modelo:

Consiste na criação de um modelo matemático abstrato que represente o problema em si, considerando as informações coletadas na etapa de formulação do problema.

### C. Otimização:

<sup>1</sup> <https://www.priberam.pt/DLPO/otimizar> [consultado em 03-05-2016].

Nesta etapa estamos interessados em obter ótimas soluções para o problema de interesse.

#### D. Implementação:

Nesta etapa as soluções obtidas devem ser testadas, comparadas e sua aplicabilidade avaliada.

As etapas de Otimização e Implementação das soluções obtidas podem fornecer informações extras que não foram consideradas nas etapas de Elaboração do Modelo. Ajustes no processo de Elaboração do Modelo e Otimização devem ser realizadas sempre que estas novas informações contribuam para a melhoria da qualidade da resposta procurada.

As etapas de Elaboração do Modelo e Otimização são fundamentais. Assim como retas e círculos, os valores que minimizam ou maximizam uma função são abstrações matemáticas: elas não existem no “mundo real”. Em nossas atividades cotidianas sempre utilizamos aproximações “boas o suficiente” de entes abstratos. O computador responsável pelo auxílio da navegação da nave Apolo 11 por exemplo, trabalhava com um tamanho de palavra de 16 bits<sup>2</sup> e executava tarefas que exigiam um alto grau de precisão, cujo fracasso traria perdas humanas, técnicas e financeiras. De maneira geral, uma solução para a formulação matemática presente na etapa de Otimização é um valor  $x^*$  suficientemente próximo do valor ótimo. Além disto, solução procurada  $x^*$  deve necessariamente satisfazer às condições que limitam o problema.

Um dos produtos da etapa de Elaboração do Modelo é a definição da função  $f(x)$ , chamada de função objetivo, que represente da melhor forma possível o problema em estudo. A complexidade e a representação da função objetivo, em conjunto com a natureza dos elementos de seu domínio determinam a técnica (ou algoritmo<sup>3</sup>) utilizada para a obtenção de uma solução na etapa de Otimização.

Os casos mais simples de função objetivo, de natureza linear, contam com métodos eficientes de resolução tais como o Simplex [111] ou o método dos pontos interiores [38]. Quando a função objetivo  $f$  é uma função diferenciável e convexa, a caracterização das condições de otimalidade [65, 7] bem como as técnicas de obtenção de soluções de problemas lineares ou não lineares unimodais estão bem solidificadas na literatura [65, 93, 6, 115, 122]. Entretanto, mesmo se a função objetivo for diferenciável, se esta for multimodal os métodos tradicionais de direção de busca ou *hill climbing* não garantem como solução um ótimo global, podendo apontar ótimos locais como solução final como ilustrado na figura 2.

Ainda em relação à função objetivo, existem outros complicadores. Ao modelarmos uma situação prática, muitas vezes a formulação matemática do problema é muito complexa,

<sup>2</sup> <http://history.nasa.gov/computers/Ch2-5.html>

<sup>3</sup> Por algoritmo se entende uma sequência finita de procedimentos que levam à solução de classes semelhantes de problemas.

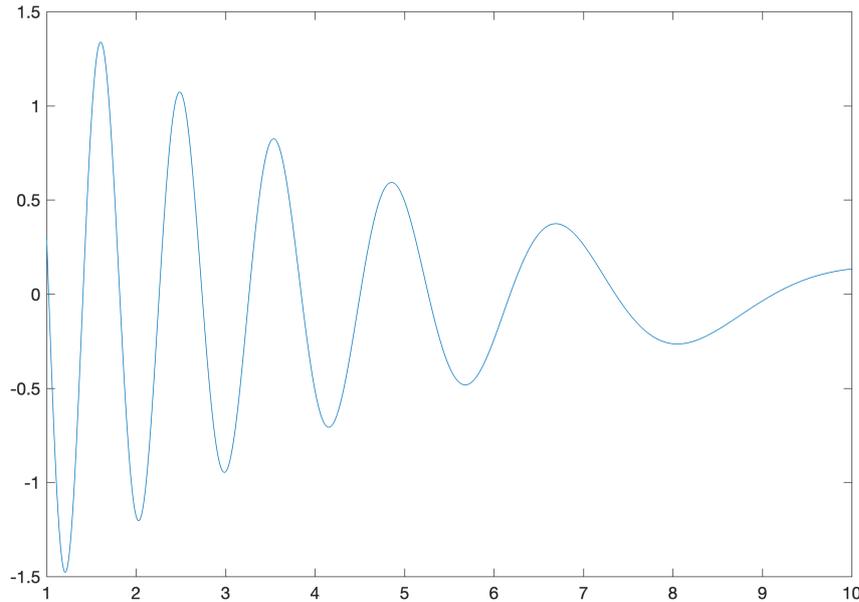


Figura 2 – A função  $y = 2 \sin(10xe^{(-0.08x)})e^{(-0.25x)}$

possui um mínimo (e um máximo) global e cinco mínimos (e quatro máximos) locais no intervalo  $(0, 10)$ . Dependendo do valor inicial, os algoritmos clássicos não convergirão para os valores extremos globais.

a ponto de não ser possível ter uma expressão analítica para a mesma. Métodos de modelagem que envolvem equações diferenciais parciais podem ser resolvidos por meio da técnica dos Elementos Finitos, que envolvem a resolução numérica de muitas equações integrais. Estes são exemplos de problemas do tipo “caixa preta” (*black box*), onde não se tem disponível nenhuma informação sobre a topologia da superfície de resposta.

Os modelos matemáticos mais simples apresentam um único objetivo a ser otimizado. Entretanto, ao elaborarmos um modelo mais sofisticado, mais de um parâmetro deve ser considerado antes da escolha da resposta. Considere por exemplo um projeto de motor. Fatores como potência, torque, consumo de combustível, confiabilidade, estimativa de vida útil, custo de produção e de manutenção são características que definem a qualidade do motor e refletem o tipo de projeto necessário para sua construção. Esta situação, onde há mais de um objetivo a ser otimizado, caracteriza a otimização multiobjetivo.

A razão da existência de algoritmos de resolução de problemas multiobjetivo é obter soluções ótimas para problemas do mundo real. Problemas artificiais utilizados para se testar a performance de algoritmos apresentam uma formulação matemática bem definida e têm sua solução ótima conhecida *a priori*. No sentido oposto, um problema real tem sua formulação matemática fortemente relacionada ao modelo elaborado para aquela situação. Um mesmo problema pode apresentar diferentes formulações matemáticas, refletindo o grau de precisão e abstração do modelo matemático associado àquele problema. A incorporação de incertezas e imprecisões, inerentes a problemas presentes em nosso cotidiano, torna o modelo mais realístico e preciso. A busca de soluções ótimas em problemas com a presença

de incertezas é classificada como otimização robusta [13, 48, 11, 42].

A dificuldade de se obter boas soluções em problemas de Otimização multiobjetivo é determinada por vários fatores, dentre eles o número de variáveis do espaço de decisão e o número de objetivos. Quando o número de objetivos é maior do que três, além da impossibilidade de visualizarmos as soluções obtidas por meio dos métodos gráficos usuais, o aumento da dimensionalidade do espaço dos objetivos reduz a eficiência dos métodos de otimização e das métricas de avaliação da qualidade das soluções obtidas, além de dificultar a escolha de uma solução em particular pelo tomador de decisão. Estas características fazem com que problemas de otimização com mais do que três objetivos recebam uma atenção especial dos pesquisadores e sejam classificados como problemas de otimização com muitos objetivos [150, 81, 10, 131].

Uma solução apresentada na literatura para superar os desafios impostos pelo aumento no número de objetivos é a exploração de regiões de interesse (Region Of Interest (ROI)) no espaço dos objetivos de problemas com muitos objetivos [82, 9]. Esta estratégia busca soluções em uma região específica do espaço dos objetivos obedecendo algum critério definido pelo tomador de decisão. Apesar desta técnica não promover a redução da dimensionalidade do problema, a restrição da busca de soluções em uma região específica aumenta a eficiência dos algoritmos de otimização e facilita a escolha das soluções finais pelo tomador de decisão, uma vez que soluções factíveis mas desinteressantes do ponto de vista do otimizador são ignoradas durante o processo de otimização.

## 1.2 Objetivos

O objetivo geral deste trabalho é estudar o problema de se obter soluções ótimas para problemas contínuos multiobjetivo e com muitos objetivos, que sejam menos sensíveis à presença de perturbações ou outros tipos de variabilidade não programada em suas variáveis. A solução deste tipo de problema faz uso dos métodos de Otimização Robusta Multiobjetivo.

Os objetivos específicos compreendem:

1. Fazer um levantamento do estado da arte dos métodos de otimização, em particular das heurísticas de otimização com inspiração evolutiva;
2. Estudar os tipos de ruídos e incertezas presentes nos processos de tomada de decisão e sua incorporação nos métodos de otimização;
3. Apresentar um novo conjunto de funções de teste para algoritmos evolutivos que incremente os desafios apresentados aos algoritmos de otimização e que seja apro-

priado à exploração de regiões de interesse do espaço dos objetivos, bem como à otimização com a presença de ruído nas variáveis do problema.

4. Apresentar uma nova métrica de avaliação de soluções que seja adequada à exploração de regiões de interesse no espaço dos objetivos;
5. Apresentar um novo método de visualização de soluções em espaços de alta dimensionalidade que seja capaz de destacar regiões específicas deste espaço, bem como distinguir soluções menos sensíveis à presença de ruído das demais soluções ótimas obtidas.
6. Apresentar um método de otimização que apresente soluções ótimas com a presença de ruído e incertezas nas variáveis de decisão da função objetivo e que seja capaz de explorar uma região de interesse específica do espaço dos objetivos.

### 1.3 Contribuições da tese

Este trabalho trata de otimização robusta em problemas multiobjetivo e com muitos objetivos. A estratégia adotada para a obtenção de soluções sub-ótimas robustas e a exploração de regiões de interesse no espaço dos objetivos. As principais contribuições deste trabalho são:

1. Um novo algoritmo evolutivo para problemas de Otimização Robusta com qualquer número de objetivos. Apesar de sua conformidade imediata com problemas práticos de engenharia, as referências à Otimização Robusta Multiobjetivo e com muitos objetivos são escassas na literatura especializada. Além disto os métodos disponíveis tem elevado custo computacional. O método apresentado neste trabalho se mostrou eficiente nos casos analisados, tanto em problemas de teste quanto em problemas práticos e apresenta baixo custo computacional, além de preencher a lacuna presente no referencial teórico.
2. Um novo conjunto de funções de teste para problemas de Otimização usual e Robusta. Em conjunto com as técnicas de Otimização Robusta Multiobjetivo, uma outra deficiência encontrada na literatura especializada foi a carência de problemas teste para Otimização Robusta Multiobjetivo. Os problemas de teste apresentados são de fácil implementação e interpretação, com Fronteira Pareto Robusta bem definida e escaláveis a qualquer número de variáveis e objetivos. Os problemas de teste apresentados neste trabalho também se aplicam à Otimização Multiobjetivo usual e trazem novos e significativos desafios aos Algoritmos de Otimização contemporâneos. Uma outra contribuição do novo conjunto de problemas de teste são as restrições de igualdade e de desigualdade que assim como a otimização robusta são tópicos pouco explorados na literatura especializada.

3. Um novo método de visualização de soluções no espaço dos objetivos para problemas de alta dimensionalidade. Por se tratar de um desafio muito grande e ter aplicação em diversas áreas, a visualização de dados em espaços com dimensão maior do que três é um tema recorrente na literatura científica. Uma vez que o suporte de exibição dos dados é sempre bidimensional (uma tela ou uma folha de papel por exemplo) a perda de informação é um fenômeno inevitável. Cada método presente na literatura enfatiza a exibição de um conjunto de características da massa de dados, omitindo outros. A ferramenta de visualização apresentada neste trabalho agrega três diferentes métodos, de modo que as informações presentes em cada um se complementam de forma harmônica e permitem a exibição de um número maior de características que um único método em particular. Além disto, a ferramenta proposta permite a análise da convergência e da robustez das soluções encontradas, além de permitir a exibição de mais de um conjunto de dados em um mesmo gráfico.
4. Uma nova metodologia de exploração de regiões de interesse no espaço dos objetivos. A dificuldade de se obter soluções de problemas de Otimização Multiobjetivo cresce muito quando o número de objetivos é maior do que três. Os problemas posicionados acima deste limiar são classificados como Problemas de Otimização com Muitos Objetivos e exigem novas abordagens para a obtenção de soluções. A estratégia adotada neste trabalho foi a exploração de Regiões de Interesse ROI no espaço dos objetivos, por preservar o número de objetivos original e permitir a incorporação de preferências do decisor no processo de otimização, além de permitir a busca de soluções robustas. Este trabalho apresenta um método inédito de determinação da ROI onde as preferências do decisor podem ser incorporadas de maneira fácil e intuitiva.
5. Uma nova métrica de avaliação da convergência para Problemas de Otimização. As métricas de avaliação de soluções obtidas por algoritmos evolutivos apresentam virtudes e defeitos. Em situações específicas, tais como problemas de otimização com muitos objetivos, a exploração de ROI e/ou a Otimização Robusta, são aplicadas as métricas usuais. Apesar de sua eficiência, estes métodos não foram especificamente projetados para estas situações. Para preencher esta lacuna na literatura científica e mensurar os resultados obtidos com mais acurácia, foi desenvolvido uma nova métrica de avaliação de resultados específica para mensurar a convergência de soluções em Regiões de Interesse de problemas com muitos Objetivos. Esta métrica foi desenvolvida tendo em mente sua aplicação em problemas de Otimização Robusta e usual. A comparação dos valores desta métrica com os resultados obtidos pelas outras métricas em problemas de otimização usual mostra a coerência dos resultados e sua superioridade nas situações para as quais ela foi projetada.

Cada um dos itens destacados acima já representam por si só um significativo avanço

nesta área de pesquisa. Todos os temas abordados são atuais e relevantes, e representam em seu conjunto uma parcela do atual desafio encontrado por muitos pesquisadores desta área. Um indicativo da importância das soluções apresentadas é, além de sua atualidade, a sua escassez na literatura especializada.

## 1.4 Organização do trabalho

Este trabalho está distribuído da seguinte maneira. Este capítulo faz a caracterização do problema de pesquisa, com uma breve apresentação das metodologias de aprimoramento da qualidade e do problema de otimização. Esta seção também apresenta o objeto de estudo do trabalho, bem como os objetivos geral e específicos que se deseja alcançar. O capítulo 2 apresenta o problema de otimização multiobjetivo e três algoritmos evolutivos consagrados na literatura. A questão do ruído e das incertezas presentes nos processos de tomada de decisão, bem como sua incorporação nos métodos de otimização é abordada no capítulo 3. O capítulo 4 faz uma revisão crítica de alguns problemas de teste apresentadas na literatura (ZDT, DTLZ e WFG) e apresenta um novo conjunto de problemas de otimização, juntamente com uma nova métrica de desempenho. O capítulo 5 faz a revisão dos principais métodos de visualização de soluções de problemas multiobjetivo e apresenta uma nova proposta, agregando três métodos de visualização de dados. O capítulo 6 é dedicado à geração de vetores de peso, que é uma das primeiras etapas dos algoritmos evolutivos que utilizam a técnica de decomposição/agregação. Neste capítulo são apresentados dois novos métodos de geração de vetores de peso: o primeiro método cria vetores de peso quase igualmente espaçados no primeiro ortante do espaço  $\mathbb{R}^M$  e o segundo método gera vetores em uma região específica deste mesmo espaço. O capítulo 7 apresenta uma nova metodologia de exploração de regiões de interesse (ROI) do espaço dos objetivos em problemas com qualquer número de objetivos, que faz uso dos cones de vetores definidos no capítulo anterior. No capítulo 8 é apresentada uma proposta de algoritmo coevolutivo competitivo para a solução do problema de otimização robusta multiobjetivo e com muitos objetivos – o CR-MOEA/D e sua versão para busca de soluções em regiões de interesse – o coneCR-MOEA/D. O capítulo 9 encerra este trabalho apresentando as conclusões e os trabalhos futuros desta pesquisa.



## 2 Otimização

### 2.1 Introdução

Esta seção trata de algumas definições e considerações gerais sobre métodos de otimização.

**Definição 1 (Problema de Otimização mono-objetivo)** *O problema de otimização mono-objetivo pode ser formulado de forma mais geral como*

$$\begin{aligned} \mathbf{x}^* &= \arg \min f(\mathbf{x}) \\ \text{sujeito a: } &\begin{cases} g_i(\mathbf{x}) \leq 0, & 1 \leq i \leq p \\ h_j(\mathbf{x}) = 0, & 1 \leq j \leq q \\ \mathbf{x} \in \mathcal{S} \end{cases} \end{aligned} \quad (2.1)$$

sendo  $g_i(\mathbf{x})$  as  $p$  restrições de desigualdade e  $h_j(\mathbf{x})$  as  $q$  restrições de igualdade.

As variáveis de decisão  $\mathbf{x} = (x_1, \dots, x_N)$  são restritas a um conjunto  $\mathcal{S}$  chamado de espaço de decisão ou espaço de busca. Os componentes  $x_i$  das variáveis de decisão podem ser números reais, números inteiros, valores booleanos ou mesmo uma mistura destes. A definição 1 apresenta um problema de minimização<sup>1</sup>. Uma solução ideal para este problema é chamada de mínimo global:

**Definição 2 (Ponto de mínimo global:)** *Seja  $f : \mathcal{S} \subset \mathbb{R} \rightarrow \mathbb{R}$  uma função real de variável real. Um mínimo global de  $f$  é um ponto  $\mathbf{x}_g \in \mathcal{S}$  tal que*

$$f(\mathbf{x}_g) \leq f(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{S} \quad (2.2)$$

Analogamente podemos definir um ponto de máximo global. De forma complementar ao ponto de mínimo global, temos os pontos de mínimos locais:

**Definição 3 (Ponto de mínimo local:)** *Seja  $f : \mathcal{S} \subset \mathbb{R} \rightarrow \mathbb{R}$  uma função real de variável real. Um mínimo local de  $f$  é um ponto  $\mathbf{x}_\ell \in I$  tal que*

$$f(\mathbf{x}_\ell) \leq f(\mathbf{x}) \quad \forall \mathbf{x} \in I \subset \mathcal{S} \quad (2.3)$$

para algum intervalo  $I = (\mathbf{x} - \epsilon, \mathbf{x} + \epsilon)$  com  $\mathbf{x} \in \mathcal{S}$ .

<sup>1</sup> A maximização de uma função  $f$  é equivalente à minimização de  $-f$ .

Os pontos de máximos locais são definidos de forma análoga. Uma função pode não ter mínimo global (ou local), bem como ter vários mínimos globais (ou locais). Por exemplo, a função  $f(\mathbf{x}) = \mathbf{x}$  não tem mínimo global/local se  $\mathcal{S} = (0, 1)$ , e na função  $f(\mathbf{x}) = 1$  todo ponto é mínimo/máximo global/local.

**Definição 4 (Problema de Otimização multiobjetivo)** *É o problema de otimização dado por*

$$\begin{aligned} \mathbf{x}^* &= \arg \min F(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_M(\mathbf{x})) \\ \text{sujeito a: } &\begin{cases} g_i(\mathbf{x}) \leq 0, & 1 \leq i \leq p \\ h_j(\mathbf{x}) = 0, & 1 \leq j \leq q \\ \mathbf{x} \in \mathcal{S} \end{cases} \end{aligned} \quad (2.4)$$

sendo  $g_i(\mathbf{x})$   $p$  restrições de desigualdade e  $h_j(\mathbf{x})$   $q$  restrições de igualdade.

Mais uma vez se considera, sem perda de generalidade, que todas as  $M$  funções  $f_i(\mathbf{x})$  devem ser minimizadas. Além do espaço de decisão  $\mathcal{S}$  (ou espaço de busca), no problema de otimização multiobjetivo temos o espaço dos objetivos, denotado por  $F$  e definido por  $F = \{\mathbf{y} \in \mathbb{R}^M, \mathbf{y} = F(\mathbf{x}) \text{ com } \mathbf{x} \in \mathcal{S}\}^2$ .

Uma característica fundamental de problemas multiobjetivo é a natureza conflitante dos objetivos envolvidos: é impossível se obter uma solução  $\mathbf{x}^*$  que minimize todos os objetivos simultaneamente. Isto acontece porque se uma variável  $\mathbf{x}$  provoca a melhoria de um objetivo  $f_i(\mathbf{x})$ , ela também provoca a deterioração de pelo menos um dos demais objetivos. Desta forma, ao contrário da otimização mono-objetivo onde existe uma relação de ordem  $\leq$  bem definida, não existe uma única solução para problemas multi-objetivo, e sim um conjunto de soluções que representa o melhor equilíbrio (*trade-off*) entre os objetivos. Para melhor caracterizar este conjunto de soluções algumas definições se fazem necessárias. Como estamos interessados na identificação de “melhores” soluções, é necessário definir o que é uma relação de ordem. Uma relação de ordem em um conjunto  $A$  é uma relação binária  $\sim$  com as seguintes propriedades [87, 22]:

**Reflexiva:**  $a \sim a$  para todo  $a \in A$ ;

**Anti-simétrica:** se  $a \sim b$  e  $b \sim a$  então  $a = b$ ;

**Transitiva:** se  $a \sim b$  e  $b \sim c$  então  $a \sim c$ .

Uma relação de ordem é total se quaisquer elementos em  $A$  podem ser comparados pela relação  $\sim$ . Caso contrário, ou seja, se nem todos os elementos de  $A$  podem ser comparados, esta é uma relação de ordem parcial. No conjunto dos números reais,  $\leq$  é

<sup>2</sup> Assumimos que cada função  $f_i(\mathbf{x})$  na definição 4 retorna um número real para cada variável  $\mathbf{x} \in \mathcal{S}$ .

uma relação de ordem total e por isto é fácil a caracterização de um mínimo global para funções  $f : \mathbb{R} \rightarrow \mathbb{R}$ . Entretanto, não existe uma relação de ordem total em  $\mathbb{R}^N$  para  $N > 1$ . Entretanto, é possível estabelecer uma relação de ordem parcial:

Considere dois vetores  $\mathbf{u} = (u_1, \dots, u_N)$  e  $\mathbf{v} = (v_1, \dots, v_N) \in \mathbb{R}^N$  e a relação  $\preceq$  definida como

$$\mathbf{u} \preceq \mathbf{v} \text{ se } u_i \leq v_i \text{ para todo } 1 \leq i \leq N \quad (2.5)$$

Se  $\mathbf{u} \preceq \mathbf{v}$  dizemos que  $\mathbf{u}$  é parcialmente menor que  $\mathbf{v}$ . É fácil ver que a relação  $\preceq$  é uma relação de ordem em  $\mathbb{R}^N$ , mas não é uma relação de ordem total, uma vez que os vetores  $\mathbf{u} = (0, 1)$  e  $\mathbf{v} = (1, 0)$  não podem ser comparados de forma alguma por meio da relação  $\preceq$ .

A caracterização das soluções do problema multiobjetivo (4) utiliza a relação de ordem parcial  $\preceq$  e é realizada por meio das seguintes definições [20]:

**Definição 5 (Dominância Pareto:)** *Sejam  $\mathbf{u} = (u_1, \dots, u_N)$  e  $\mathbf{v} = (v_1, \dots, v_N)$  pontos factíveis do problema de otimização multiobjetivo 4. Se  $F(\mathbf{u}) \preceq F(\mathbf{v})$  e  $F(\mathbf{u}) \neq F(\mathbf{v})$  então dizemos que  $\mathbf{u}$  domina  $\mathbf{v}$  e escrevemos  $\mathbf{u} \prec \mathbf{v}$ .*

Caso não haja ambiguidade, é usual dizer que se  $F(\mathbf{u}) \preceq F(\mathbf{v})$  e  $F(\mathbf{u}) \neq F(\mathbf{v})$  então  $F(\mathbf{u})$  domina  $F(\mathbf{v})$ . Como  $\preceq$  é uma relação de ordem parcial, existem pontos de  $\mathbb{R}^N$  que não podem ser comparados por esta relação de ordem. Se, para dois pontos distintos  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^N$  tivermos que as afirmações  $\mathbf{u} \prec \mathbf{v}$  e  $\mathbf{v} \prec \mathbf{u}$  são ambas falsas, ou seja,  $\mathbf{u}$  não domina  $\mathbf{v}$  e  $\mathbf{v}$  não domina  $\mathbf{u}$ , então  $\mathbf{u}$  e  $\mathbf{v}$  são não dominados e escrevemos  $\mathbf{u} \not\prec \mathbf{v}$ .

**Definição 6 (Conjunto Pareto Ótimo:)** *Seja  $F$  a função objetivo de um problema multiobjetivo. O conjunto*

$$P = \{\mathbf{x} \in \mathcal{S}, \nexists \mathbf{y} \in \mathcal{S} \text{ tal que } F(\mathbf{y}) \prec F(\mathbf{x})\} \quad (2.6)$$

*é chamado de Conjunto Pareto Ótimo.*

Os elementos do conjunto  $P$  são chamados de soluções eficientes do problema de otimização multiobjetivo e é este conjunto de pontos que buscamos como solução do problema (4). É uma consequência imediata da definição de  $P$  que quaisquer dois de seus elementos são não dominados, ou seja, se  $\mathbf{u}, \mathbf{v} \in P$  então  $\mathbf{u} \not\prec \mathbf{v}$ . Associado a este conjunto está a Fronteira Pareto, denotada por  $FP$  (ou Pareto Front (PF)) e definida como a imagem no espaço dos objetivos das soluções eficientes, ou seja  $FP = F(P)$ . Os elementos que compõem a solução de um problema de otimização multiobjetivo são ilustrados na Figura 3.

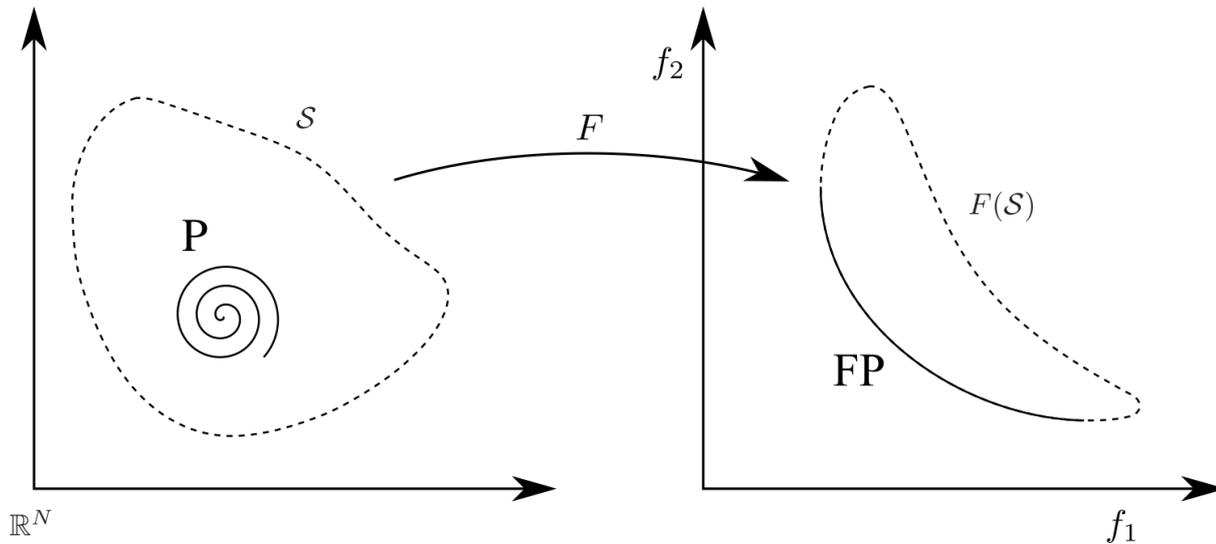


Figura 3 – Um problema de otimização multiobjetivo com dois objetivos  $f_1$  e  $f_2$  a serem minimizados, sem restrições. Estão destacados o conjunto de pontos factíveis  $\mathcal{S}$  no espaço de decisão  $\mathbb{R}^N$ , a imagem do conjunto dos pontos factíveis  $F(\mathcal{S})$  no espaço dos objetivos  $\mathbb{R}^2$ , o conjunto Pareto Ótimo  $P \subset \mathcal{S}$  e a Fronteira Pareto  $FP \subset F(\mathcal{S})$ .

## 2.2 Algoritmos Evolutivos

O termo heurística<sup>3</sup> é usado como o conjunto de técnicas utilizadas para se resolver um problema. No contexto dos problemas de otimização, uma heurística (ou meta-heurística<sup>4</sup>) é um algoritmo que busca “boas soluções” para problemas de larga escala em um tempo de execução “razoável”. Em uma heurística não existe a garantia de que a solução obtida seja ótima nem que esteja próxima da solução real, mas a solução apresentada é a melhor quando comparada a uma grande variedade de respostas possíveis para o problema [22, 140, 20].

De maneira geral, o que uma heurística faz é a exploração do espaço de busca procurando “boas soluções” para um problema. A busca por uma solução ótima pode ser feita por meio de uma única solução candidata, como nas heurísticas de busca local e recozimento simulado (*simulated annealing*), ou por meio de um conjunto de soluções candidatas (chamadas de população) como nas heurísticas de nuvem de partículas (*particle swarm*) ou nos algoritmos evolutivos. Já a exploração do espaço de busca pode ser feita por meio de decisões determinísticas, onde uma nova solução é gerada a partir de uma solução existente por meio de regras bem definidas (busca local, busca tabu) ou de forma estocástica, onde a escolha de novas soluções é influenciada por fatores aleatórios. Em uma heurística determinística, valores iniciais iguais levam à mesma solução. Já em uma heurística estocástica, um valor inicial pode levar a diferentes soluções. Uma outra

<sup>3</sup> Do grego *heuriskein* (*εὐρησκή*): descobrir novos caminhos.

<sup>4</sup> Termo introduzido por Glover [45] ao se referir a heurísticas “superiores” (como por exemplo, a Busca Tabu) quando comparadas com outras. Este texto não fará distinção entre estes dois termos.

classificação para as heurísticas pode ser feita no que diz respeito a sua estrutura. Há heurísticas inspiradas em fatores biológicos (tais como os algoritmos evolutivos, sistemas imunológicos), em fatores sociais (colônias de formigas e de abelhas, nuvem de partículas) e em fenômenos físicos (*simulated annealing*), dentre outros.

A exploração de espaços de busca de grande porte acoplados a funções sofisticadas sempre foi um grande desafio, bem como a tentativa de se produzir programas de computador dotados de inteligência e poder de decisão autônoma. Uma solução para este problema é a inspiração em processos físicos, biológicos e sociais para estes desafios. Holland [59] apresenta uma sólida fundamentação teórica de como uma abstração da evolução biológica proposta por Darwin pode ser aplicado na resolução deste tipo de problema.

As heurísticas inspiradas nos processos de seleção natural são denominados de Algoritmos Evolutivos, sendo o algoritmo genético [47, 96, 5, 26] o exemplo mais proeminente. O Algoritmo Genético é uma heurística onde um conjunto de soluções candidatas (chamado de população) faz a busca no espaço de decisão gerando novas soluções por meio de dois mecanismos de natureza estocástica (a recombinação ou *crossover* e a mutação) em um número limitado de iterações, sendo que cada iteração é denominada de geração. Em cada iteração, os elementos da população (chamados de indivíduos) recebem um valor de aptidão (*fitness*) que mede seu desempenho em relação aos demais diante do problema proposto ao algoritmo. Em seguida, os elementos de melhor aptidão são selecionados para a geração seguinte, mantendo-se fixo o tamanho da população. O Algoritmo 1 apresenta a estrutura de um algoritmo genético básico.

---

**Algoritmo 1:** Algoritmo Genético básico

---

```

1 begin
2    $t \leftarrow 0$ ;
3    $x_t \leftarrow$  população inicial;
4    $f_t(\mathbf{x}) \leftarrow$  avaliação da aptidão;
5   while critério de parada = falso do
6      $x_t^s \leftarrow$  seleção ( $x_t$ );
7      $x_t^c \leftarrow$  recombinação ( $x_t^s$ );
8      $x_t^e \leftarrow$  mutação ( $x_t^c$ );
9      $f_t(\mathbf{x}^e) \leftarrow$  avaliação da aptidão ( $x_t^e$ );
10     $x_t \leftarrow$  atualização da população ( $x_t, x_t^e$ );
11     $t \leftarrow t + 1$ ;
12  end while
13 end

```

---

As principais etapas de um Algoritmo Genético são apresentadas a seguir, supondo que este será aplicado na minimização de um único objetivo  $F : \mathcal{S} \subset \mathbb{R}^N \rightarrow \mathbb{R}$  onde  $\mathcal{S}$  é um hipercubo em  $\mathbb{R}^N$ , ou seja, se  $\mathbf{x} \in \mathcal{S}$  então  $\mathbf{x} = (x_1, \dots, x_N)$  com  $\ell_i \leq x_i \leq u_i$  sendo  $\ell_i$  e  $u_i$  respectivamente os limites inferior e superior de cada coordenada do vetor  $\mathbf{x}$ .

**Inicialização das variáveis:** Seguindo a analogia com processos biológicos, o conjunto soluções candidatas é chamado de população e cada solução candidata é chamada de indivíduo. Nas propostas originais de Holland e Goldberg, o vetor de variáveis é denotado por cromossomo, cada componente é chamada de **gene** e seu valor é o alelo e as variáveis apresentavam a codificação binária de modo a ser mais fiel à inspiração biológica da heurística. Entretanto a codificação por meio de números na base dez (chamada de codificação real) é muito utilizada<sup>5</sup>. De maneira geral, define-se o tamanho  $P$  da população e se inicializam as variáveis  $\mathbf{x} = (x_1, \dots, x_N)$  seguindo uma distribuição uniforme, ou seja,  $x_i \in U[\ell_i, u_i]$ . Maaranen et al. [94] comparam diferentes formas de inicialização das variáveis em Algoritmos Genéticos, procurando uma maior dispersão dos pontos no espaço de busca.

**Avaliação da aptidão:** Ou *fitness* deve assumir a forma mais adequada para o problema. Pode-se utilizar o valor nominal da função  $f(\mathbf{x})$  ou procurar normalizações de modo que  $0 \leq f(\mathbf{x}) \leq 1 \forall \mathbf{x} \in \mathcal{S}$ . Uma técnica muito comum em problemas com restrições é a incorporação das mesmas como uma penalidade da função objetivo.

**Seleção:** A criação de novas soluções candidatas tem início no processo de seleção onde, de forma aleatória, pontos com bons valores de função de aptidão são escolhidos. Este processo não pode se resumir aos indivíduos com melhor valor de aptidão a fim de se evitar a convergência prematura em um mínimo local. Pelo mesmo motivo, indivíduos com valores ruins de aptidão também não devem ser ignorados. A forma mais simples de seleção é o torneio binário: aleatoriamente selecionam-se dois indivíduos e é escolhido aquele de melhor valor de função de aptidão. Esta forma de seleção pode ser estendida a um torneio com  $k$  indivíduos. Outra forma clássica de seleção é por meio do valor relativo de sua função de aptidão  $f_i$  de modo que a probabilidade de seleção do  $i$ -ésimo indivíduo é dada por

$$p_i = \frac{f_i}{\sum_{j=1}^k f_j}$$

Considerando a função de distribuição acumulada  $P_i = \sum_{j=1}^i p_j$ , associa-se o  $i$ -ésimo indivíduo  $\mathbf{x}^i$  da população ao intervalo  $(P_{i-1}, P_i]$ , sendo  $P_0 = 0$ . Temos dois métodos de seleção usuais:

**Roleta (*roulette-wheel*):** Para a escolha de  $k$  indivíduos, geram-se aleatoriamente  $k$  números  $r_i$  com  $r_i \in \mathcal{U}[0, 1]$ ,  $r_i \neq 0$ . O indivíduo  $\mathbf{x}^j$  é selecionado se  $r_i \in (P_{j-1}, P_j]$ .

<sup>5</sup> A codificação das variáveis também deve se adequar ao tipo do problema. Por exemplo, no problema da mochila (*knapsack problem*), a codificação binária é mais adequada.

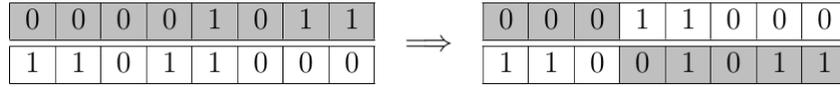


Figura 4 – Recombinação binária de um ponto

**SUS (Stochastic Universal Sampling):** Para a escolha de  $k$  indivíduos, gera-se aleatoriamente um número  $r \in \mathcal{U}[0, 1]$ ,  $r \neq 0$ , e em seguida consideram-se  $k$  números igualmente espaçados dados por

$$R = \left\{ r, r + \frac{1}{k}, r + \frac{2}{k}, \dots, r + \frac{k-1}{k} \right\} \text{ mod } 1.$$

O indivíduo  $\mathbf{x}^j$  é selecionado se  $(r + \frac{\alpha}{k}) \text{ mod } 1 \in (P_{j-1}, P_j]$  com  $0 \leq \alpha \leq k-1$ .

**Recombinação:** Originalmente chamado de *crossover*, é o operador responsável pela geração de novas soluções candidatas. Estas novas soluções são também chamadas de *offspring*. São selecionados dois vetores representativos da população e uma nova solução candidata é criada. Procura-se criar novas soluções que tenham alguma relação de proximidade com as soluções previamente selecionadas e desta maneira fazer a exploração do espaço de busca.

No caso da representação binária, é escolhido aleatoriamente um ponto de corte em uma posição dos vetores e um novo vetor é criado pela combinação destas partes (veja Figura 4). É possível generalizar esta técnica de recombinação para  $k$  pontos de corte.

Quando se utiliza a codificação real, a recombinação mais simples utiliza uma combinação convexa da forma  $\mathbf{y} = t\mathbf{x}^a + (1-t)\mathbf{x}^b$ , sendo  $\mathbf{x}^a$ ,  $\mathbf{x}^b$  os elementos da população previamente selecionados,  $\mathbf{y}$  o novo ponto criado e  $t \in U[0, 1]$ .

Um método de recombinação muito utilizado é o SBX (*Simulated Binary Crossover* – [28]), que simula a recombinação com um ponto de corte para vetores com codificação binária. A recombinação SBX cria dois novos indivíduos  $\mathbf{y}^a$  e  $\mathbf{y}^b$  a partir de dois elementos previamente selecionados  $\mathbf{x}^a = (x_1^a, \dots, x_N^a)$  e  $\mathbf{x}^b = (x_1^b, \dots, x_N^b)$  da seguinte forma:

1. Para cada  $i \in \{1, \dots, N\}$  calcule

$$\beta_i = \begin{cases} \frac{1}{(2u_i)\eta_c + 1} & \text{se } u_i \leq 0.5 \\ \left( \frac{1}{2(1-u_i)} \right) \frac{1}{\eta_c + 1} & \text{caso contrário} \end{cases}$$

sendo  $u_i \in U[0, 1]$  e  $\eta_c$  um parâmetro definido pelo tomador de decisão que controla a “proximidade” com que as novas soluções candidatas estarão dos

indivíduos previamente selecionados. Quanto maior o valor de  $\eta_c$ , maior a probabilidade do novo elemento estar próximo do antigo.

2. Gere dois novos indivíduos  $\mathbf{y}^a = (y_1^a, \dots, y_n^a)$  e  $\mathbf{y}^b = (y_1^b, \dots, y_n^b)$  por meio das expressões abaixo:

$$\begin{aligned} y_i^a &= 0.5 [(1 + \beta_i)x_i^a + (1 - \beta_i)x_i^b] \\ y_i^b &= 0.5 [(1 - \beta_i)x_i^a + (1 + \beta_i)x_i^b] \end{aligned}$$

Independente da escolha do método de recombinação, forma de criação de novas soluções candidatas ocorre com uma probabilidade de ocorrência  $\rho_c$ .

**Mutação:** Após a criação de uma nova solução candidata, esta pode sofrer uma pequena perturbação procurando maximizar a exploração do espaço de busca. Este operador ocorre com uma probabilidade pequena  $\rho_m \ll 1$  e sua forma mais simples pode ser descrita como  $\mathbf{y} = \mathbf{x} + \tau$ , sendo  $\mathbf{y}$  o novo ponto a ser gerado,  $\mathbf{x}$  o ponto alvo da mutação e  $\tau$  uma perturbação seguindo alguma distribuição de probabilidade (uniforme, normal, etc).

Com a mesma inspiração da recombinação SBX, Deb and Goyal [29] sugerem a mutação polinomial onde um novo vetor  $\mathbf{y} = (y_1, \dots, y_n)$  é gerado a partir de um ponto  $\mathbf{x} = (x_1, \dots, x_n)$  previamente selecionado. A  $i$ -ésima coordenada do novo ponto é dada por  $y_i = x_i + \delta_i(x_i^u - x_i^l)$ , onde  $x_i^u$  e  $x_i^l$  correspondem aos limites superior e inferior da  $i$ -ésima coordenada do ponto  $\mathbf{x}$  e

$$\delta_i = \begin{cases} \frac{1}{(2u_i)^{\eta_m + 1} - 1} & \text{se } u_i < 0.5 \\ 1 - \frac{1}{[2(1 - u_i)]^{\eta_m + 1}} & \text{caso contrário} \end{cases}$$

onde  $u_i \in U[0, 1]$  e  $\eta_m$  um parâmetro definido pelo Tomador de Decisão. Mais uma vez, quanto maior o valor de  $\eta_m$ , maior a probabilidade do novo elemento estar próximo do antigo.

A atualização da população pode seguir as mesmas regras da etapa de seleção. Para a otimização multiobjetivo, os mesmos passos podem ser seguidos, exceto a atualização da população. Na otimização mono-objetivo a convergência da população em torno do ótimo global não prejudica a qualidade da solução, apesar de não ser desejável. Entretanto, na otimização multiobjetivo, um indicador de qualidade de uma solução é sua dispersão na Fronteira Pareto, de modo que qualquer técnica de otimização multiobjetivo tem sempre dois desafios: a convergência da população na Fronteira Pareto e sua dispersão ao longo da mesma.

## 2.2.1 Técnicas avançadas em Computação evolutiva

### 2.2.1.1 Nicho

A intensa utilização de algoritmos evolutivos em problemas de otimização faz com que muitos pesquisadores aprimorem as técnicas básicas. Uma situação muito comum que deve ser evitada é a convergência prematura da população em torno de um ótimo local, provocada por métodos de seleção demasiadamente elitistas, que faz com que o algoritmo perca sua capacidade de exploração do espaço de busca. Técnicas de nicho (*Niching methos* [95, 129]) incorporadas a Algoritmos Genéticos permitem a manutenção da diversidade da população evitando a convergência prematura. Uma técnica de nicho muito utilizada é a *fitness sharing*. Para um indivíduo com *fitness*  $f_i$ , sua função de *fitness sharing*  $f'_i$  é dada por

$$f'_i = \frac{f_i}{m_i}$$

sendo  $m_i$  uma medida de proximidade do  $i$ -ésimo elemento com o restante da população calculada como

$$m_i = \sum_{j=1}^k sh(d_{i,j})$$

sendo  $k$  o tamanho da população,  $d_{i,j}$  a distância entre os elementos  $i$  e  $j$  e  $sh(d_{i,j})$  uma função que mede o nível de similaridade entre elementos sa população. A expressão mais utilizada para esta função de similaridade é dada por

$$sh(d_{i,j}) = \begin{cases} 1 - \left(\frac{d_{i,j}}{\sigma_s}\right)^\alpha, & \text{se } d_{i,j} < \sigma_s \\ 0 & \text{caso contrário} \end{cases} \quad (2.7)$$

onde  $\sigma_s$  é uma distância limite para a medição de dissimilaridade e  $\alpha$  um parâmetro que determina o formato desta função. Uma outra forma de preservar a diversidade da população é a distância de multidão (*crowding distance*), proposta por Deb et al. [34].

### 2.2.1.2 Autoadaptação

Um aspecto presente em sistemas biológicos é a autoadaptação, que pode ser transposta para algoritmos evolutivos por meio de um operador de mutação dinâmico [1, 5]. Para um vetor  $\mathbf{x}^{(t)} = (x_1^{(t)}, \dots, x_n^{(t)})$  em uma geração  $t$ , esta estratégia de mutação consiste na criação de um novo ponto  $\mathbf{x}^{t+1}$  seguindo a seguinte regra:

$$\begin{aligned} \sigma^{(t+1)} &= \sigma^{(t)} \cdot \exp(\tau \cdot N(0, 1)), \quad \tau = n^{-1/2} \\ x_i^{(t+1)} &= x_i^{(t)} + \sigma^{(t+1)} \cdot N_i(0, 1) \end{aligned}$$

onde o fator de dispersão  $\sigma t$  é atualizado a cada geração.

### 2.2.1.3 Coevolução

Outro elemento presente em sistemas biológicos é a coevolução. Um exemplo de coevolução (social) se dá pela interação simbiótica entre homens e cães, presente desde as primeiras comunidades de humanos caçadores-coletores. Os cães alertavam a presença de intrusos e auxiliavam nas caçadas. Em troca, recebiam comida e proteção. Esta parceria nunca se rompeu e hoje a população de cães domésticos é maior do que a de lobos e cães selvagens [54]. Coello et al. [20] destacam dois tipos principais de algoritmos coevolutivos:

**Coevolução competitiva:** Neste cenário, a função de aptidão de um indivíduo é o resultado de uma série de “encontros” com outros indivíduos da mesma população, sendo comum sua aplicação em jogos.

**Coevolução cooperativa:** A função de aptidão de um indivíduo recebe a influência de indivíduos de outras espécies (outras populações).

Sendo a estratégia mais utilizada para a resolução de problemas de otimização. Rosin and Belew [123, 124] e Stanley and Miikkulainen [136] consideram a coevolução competitiva como uma evolução simultânea entre duas ou mais populações, onde a aptidão de um indivíduo em uma população “hospedeira” tem seu valor definido por meio de uma competição com outro indivíduo em uma população de “parasitas”. Nestes trabalhos, os autores consideram a interação de uma população de “parasitas” para a formação da função de aptidão (utilizando técnicas de *fitness sharing*) dos indivíduos na população de “hospedeiros”. Rosin and Belew [123, 124] utilizam esta estratégia em um Algoritmo Genético aplicados a jogos (Tic-Tac-Toe, Nim e Go), enquanto que Stanley and Miikkulainen [136] tratam de movimento de robôs em uma competição<sup>6</sup>, também utilizando um Algoritmo Genético.

### 2.2.1.4 Decomposição/Agregação

Uma técnica aplicada a algoritmos genéticos é a estratégia da decomposição [97], que consiste na divisão do problema principal em subproblemas de menor complexidade. Esta técnica é muito aplicada na otimização multiobjetivo [169, 31, 3] e também em problemas de otimização mono-objetivo [132]. Sua implementação consiste na definição *a priori* de um conjunto de direções de busca no espaço dos objetivos dado pelos vetores  $W = \{\mathbf{w}^1, \dots, \mathbf{w}^k\}$  e sua associação a pontos do espaço de busca. Cada subproblema se resume na aplicação de alguma técnica de agregação [151] (soma ponderada, norma de Tchebycheff ou *Boundary Intersection* - BI) aplicada a um conjunto de pontos no espaço de objetivos, que por sua vez estão associados a direções de busca  $\mathbf{w}^i \in W$ .

<sup>6</sup> Mais detalhes sobre o movimento dos robôs estão disponíveis em [www.cs.utexas.edu/users/nn/pages/research/neatdemo.html](http://www.cs.utexas.edu/users/nn/pages/research/neatdemo.html).

### 2.2.2 MOEA/D

Um algoritmo que utiliza a técnica de decomposição é o MOEA/D proposto por Zhang and Li [166]. Nesta heurística, um conjunto de direções de busca  $W = \{\mathbf{w}, \dots, \mathbf{w}_P\}$  é criado no espaço dos objetivos com a mesma quantidade de vetores ( $P$  vetores) da população  $X$  e é realizada uma associação um-a-um entre as soluções candidatas e os elementos de  $W$ : cada elemento de  $W$  está associado a um único elemento da população e vice-versa. Os vetores em  $W$  orientam o processo de busca promovendo a decomposição do problema original dado por  $F(\mathbf{x}) = (f_1, \dots, f_M)$  em  $P$  subproblemas utilizando alguma técnica de agregação. Cada subproblema considera a minimização simultânea dos  $M$  objetivos em uma direção do espaço dos objetivos dada por  $\mathbf{w}^i \in W$ , utilizando uma subpopulação  $X_{\mathbf{w}^i}$  da população original. Os elementos da subpopulação  $X_{\mathbf{w}^i}$  são definidos por uma relação de proximidade ( $k$ -vizinhos próximos) de sua imagem no espaço dos objetivos em relação a  $\mathbf{w}^i$ . No final de cada geração, as melhores soluções são armazenadas em uma população externa  $EP$ . Alguns métodos fazem uso a solução utópica (também chamada de ponto ideal)  $\mathbf{z} = (z_1, \dots, z_M)$  sendo  $z_i = \min f_i(\mathbf{x})$  no problema de minimização. Os seguintes métodos de agregação são utilizados:

**Soma Ponderada (*Weighted Sum*):** Considera a combinação convexa dos  $M$  objetivos.

Para um vetor  $\mathbf{w} = \{w_1, \dots, w_M\}$ , com  $\sum_{i=1}^M w_i = 1$ , este método considera o subproblema

$$\text{minimize } g^{ws}(\mathbf{x}|\mathbf{w}) = \sum_{i=1}^M w_i f_i(\mathbf{x}) \quad (2.8)$$

onde  $\mathbf{x}$  é um ponto factível do espaço de busca  $\mathcal{S}$ . Esta estratégia tem sucesso no caso da Fronteira Pareto ser convexa, não apresentando uma boa dispersão caso contrário.

**Distância de Tchebycheff ponderada:** Utiliza na definição do problema escalar a distância de Tchebycheff entre  $f(\mathbf{x})$  e o ponto ideal  $\mathbf{z}$ , ponderada por  $\mathbf{w}_i$ :

$$\text{minimize } g^{tche}(\mathbf{x}|\mathbf{w}, \mathbf{z}) = \max_{1 \leq i \leq m} \{w_i |f_i(\mathbf{x}) - z_i|\} \quad (2.9)$$

onde  $\mathbf{x}$  é um ponto factível do espaço de busca  $\mathcal{S}$ .

**Penalty Boundary Intersection:** Esta técnica é uma adaptação da proposta de Das and Dennis [24] (*Normal Boundary Intersection - NBI*). O subproblema a ser minimizado consiste na soma penalizada (por  $\theta$ ) entre a norma da projeção ortogonal

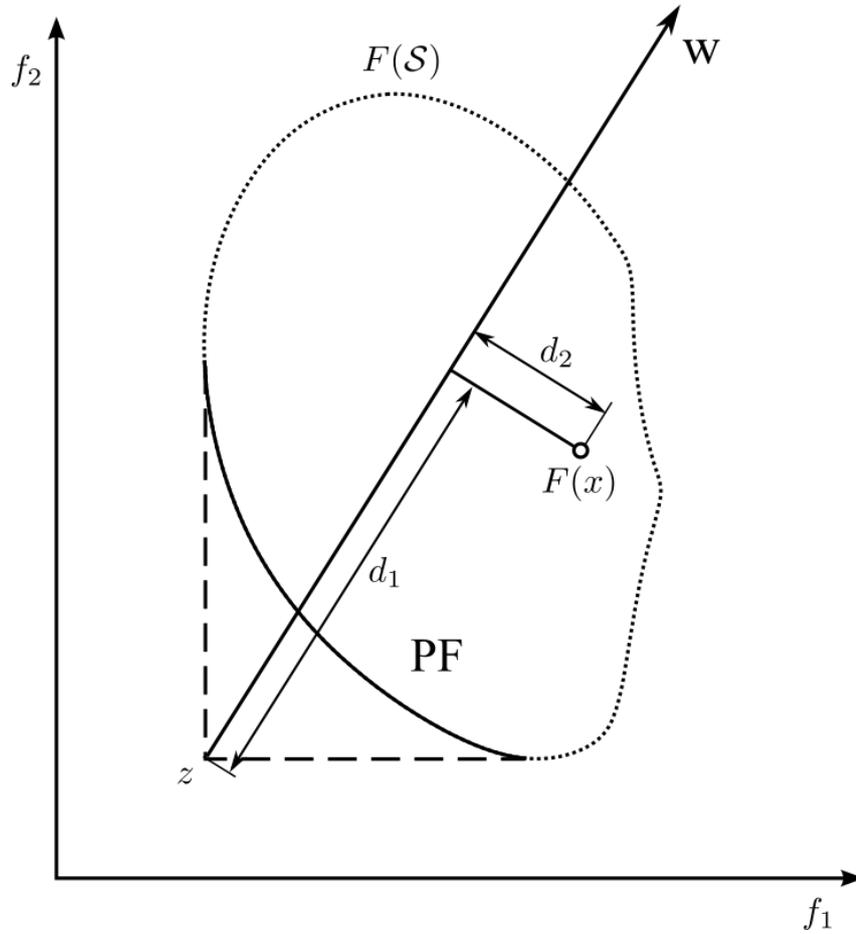


Figura 5 – Elementos do método de agregação dos objetivos *Penalty Boundary Intersection* (PBI)

de  $(f(\mathbf{x}) - \mathbf{z})$  em  $\mathbf{w}$  ( $d_1$ ) e a distância entre  $f(\mathbf{x})$  e  $\mathbf{w}$  ( $d_2$ ). Estes elementos estão ilustrados na Figura 5 e é definida por

$$\text{minimize } g^{pbi}(\mathbf{x}, \mathbf{w}, \mathbf{z}) = d_1 + \theta d_2 \quad (2.10)$$

onde  $\mathbf{x}$  é um ponto factível do espaço de busca  $\mathcal{S}$  e

$$d_1 = \frac{\|(F(\mathbf{x}) - \mathbf{z})^t \mathbf{w}\|}{\|\mathbf{w}\|}$$

$$d_2 = \left\| F(\mathbf{x}) - \left( \mathbf{z} + d_1 \frac{\mathbf{w}}{\|\mathbf{w}\|} \right) \right\|$$

sendo  $(\cdot)^t$  a operação de transposição de vetores e  $\|\cdot\|$  sua norma.

Os detalhes deste algoritmo são apresentados a seguir, considerando o método de agregação da minimização da distância de Tchebycheff ponderada  $g^{tche}(\mathbf{x}|\mathbf{w}, \mathbf{z})$ .

**Inicialização:** Após se definir o número de subproblemas  $P$ , o tamanho da subpopulação  $T$  e a criação do conjunto de direções de busca  $W = \{\mathbf{w}^1, \dots, \mathbf{w}^P\}$ , calcule a distância  $d_{i,j}$  entre dois elementos  $\mathbf{w}^i$  e  $\mathbf{w}^j$  quaisquer de  $W$ . Para cada  $i = 1, \dots, P$  crie o conjunto de índices  $B(i) = \{i_1, \dots, i_T\}$  composto pelos  $T$  vetores  $\mathbf{w}_1^i, \dots, \mathbf{w}_T^i$  mais próximos de  $\mathbf{w}^i$ . Gere a população inicial  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_P\}$  e seja  $FV^i = F(\mathbf{x}_i)$ . Crie o ponto de referência (solução ideal)  $\mathbf{z} = \{z_1, \dots, z_m\}$ , onde  $z_i = \min f_i(\mathbf{x}_j)$  com  $\mathbf{x}_j \in X$  e crie uma associação entre os elementos de  $X$  e  $W$ .

**Ciclo Evolucionário:** Até que algum critério de parada seja alcançado, a cada geração repita os procedimentos abaixo para cada elemento  $\mathbf{x}^i$  da população. Cada sequência de procedimentos listado a seguir corresponde a um ciclo completo de otimização de um dos  $P$  subproblemas:

1. **Seleção e Reprodução:** Aleatoriamente selecione dois índices  $k, l$  de  $B(i)$  e gere uma nova solução  $\mathbf{y}$  a partir de  $\mathbf{x}^k$  e  $\mathbf{x}^l$  por meio dos operadores genéticos de recombinação e mutação e verifique se o novo indivíduo é factível, ou seja se  $\mathbf{y} \in \mathcal{S}$ .
2. **Atualização do ponto de referência  $\mathbf{z}$ :** Calcule  $F(\mathbf{y}) = (f_1(\mathbf{y}), \dots, f_m(\mathbf{y}))$ . Se  $f_i(\mathbf{y}) < z_i$  faça  $z_i = f_i(\mathbf{y})$ .
3. **Atualização da vizinhança:** Para  $j \in B(i)$  se  $g^{tche}(\mathbf{y}|\mathbf{w}^j, \mathbf{z}) < g^{tche}(\mathbf{x}^j|\mathbf{w}^j, \mathbf{z})$  faça  $\mathbf{x}^j = \mathbf{y}$  e  $FV^j = F(\mathbf{y})$ .
4. **Atualização da população externa:** Remova os pontos dominados por  $F(\mathbf{y})$ . Insira  $F(\mathbf{y})$  se não houver elementos em  $EP$  que o dominem.

O algoritmo MOEA/D é responsável pela popularização do uso das técnicas de decomposição e agregação em problemas de otimização multiobjetivo. Devido a sua versatilidade, muitas variações deste algoritmo estão presentes na literatura especializada. Trivedi et al. [146] e Wang et al. [155] apresentam algumas modificações neste algoritmo visando à melhoria de sua performance. Dentre as principais modificações, destacamos as seguintes:

- **Ponto de referencia penalizado:** Na proposta original do algoritmo MOEA/D, o ponto de referencia  $\mathbf{z} = (z_1, \dots, z_M)$  utilizado nas funções de agregação em problemas de minimização é definido como sendo o valor mínimo de cada objetivo na iteração corrente, ou seja,  $z_i = \min f_i(\mathbf{x})$ ,  $0 \leq i \leq M$ . Entretanto, o uso de uma penalidade  $\epsilon$  neste ponto provoca uma melhoria na performance do algoritmo. Com esta proposta, o novo ponto de referencia é definido como  $\mathbf{z}^* = \mathbf{z} - \epsilon$ .
- **Seleção para reprodução:** Na proposta original do MOEA/D, apenas elementos da subpopulação  $B(i)$  associados ao elemento  $\mathbf{x}^i$  eram selecionados para a produção

**Algoritmo 2:** MOEA/D

---

```

input : número de subproblemas  $P$ , tamanho da subpopulação  $T$ 
output : População externa  $EP$ 
1 begin
2    $t \leftarrow 0$ ;
3    $\mathbf{w} \leftarrow$  Direções de busca ( $P$ );
4    $B(i) \leftarrow$  Estrutura de vizinhança ( $T$ );
5    $X \leftarrow$  população inicial ( $P$ );
6    $FV(X) \leftarrow$  avaliação da aptidão ( $X$ );
7    $\mathbf{z} \leftarrow$  ponto de referência ( $FV(X)$ );
8    $EP = \emptyset$  População externa;
9   while critério de parada = falso do
10    for  $i=1, \dots, P$  do
11       $k, l \leftarrow$  seleção ( $B(i)$ );
12       $\mathbf{y} \leftarrow$  operadores genéticos ( $\mathbf{x}^k, \mathbf{x}^l$ );
13       $\mathbf{z} \leftarrow$  atualização dos ponto de referência ( $FV(\mathbf{y})$ );
14      for  $j \in B(i)$  do
15         $X, FV \leftarrow$  atualização da vizinhança ( $g^{tche}(\mathbf{y}|\mathbf{w}^j, \mathbf{z}), g^{tche}(\mathbf{x}^j|\mathbf{w}^j, \mathbf{z})$ );
16      end for
17       $EP \leftarrow$  atualização da população externa  $FV$ ;
18    end for
19     $t \leftarrow t + 1$ ;
20  end while
21 end

```

---

de novos indivíduos. A fim de maximizar a exploração do espaço de busca, indivíduos da população que não estão associados ao conjunto  $B(i)$  são selecionados segundo uma baixa probabilidade  $\mu_i$ .

- **Reinserção de novos elementos:** Na proposta original do MOEA/D um novo indivíduo com melhor avaliação pela função de agregação sempre substitui o elemento ao qual foi comparado na vizinhança  $B(i)$ , o que pode provocar a deterioração da diversidade da população. Uma forma de evitar este efeito é limitar esta substituição a um número máximo de indivíduos.

### 2.2.3 NSGA-III

Um outro algoritmo que utiliza a estratégia de decomposição é o NSGA-III. Proposto por Deb and Jain [31] e Seada and Deb [132] como uma evolução do NSGA-II [34] por utilizar seu método de particionar a população em *ranks* de dominação (*Non Dominated Sorting*). O NSGA-III difere do NSGA-II por utilizar o conjunto de direções de busca  $\mathbf{w} = \{\mathbf{w}^1, \dots, \mathbf{w}^P\}$  normalizados, ou seja,  $\sum_{i=1}^M w_i^j = 1$  para todo  $j = 1, \dots, P$ . Outra diferença entre estes algoritmos está no procedimento de atualização da população entre uma geração e outra, substituindo a distância de multidão (*Crowding Distance*) como

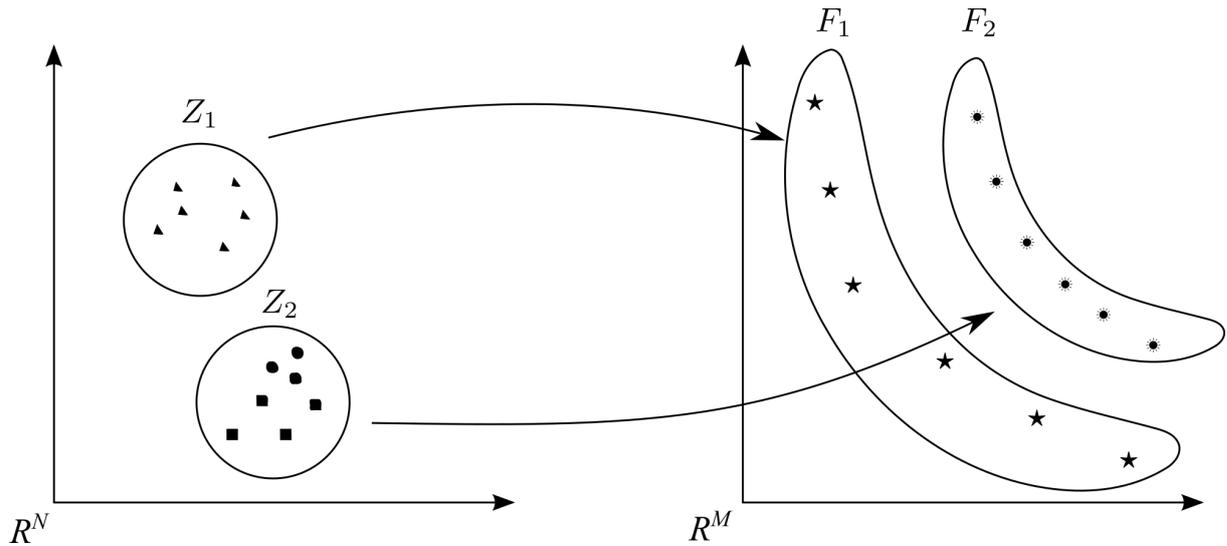


Figura 6 – Os níveis de dominância  $F_1$  e  $F_2$

mecanismo de manutenção da diversidade da população por outro procedimento que considera a distribuição da população nos pontos de referência.

Considerando  $q$  divisões em cada eixo do espaço  $M$ -dimensional dos objetivos ( $M$  objetivos), o número de pontos de referência  $H$  criados é dado por  $H = \binom{M+q-1}{q}$ . O tamanho da população  $P$  é o menor múltiplo de quatro maior do que  $H$  de modo que cada elemento da população  $\mathbf{x}$  esteja associado a uma direção de busca. No ciclo evolucionário, dois elementos da população são aleatoriamente selecionados para a aplicação dos operadores genéticos de recombinação (SBX) e mutação, gerando uma nova população  $Y$  de tamanho  $p$ . Em seguida estas duas populações são combinadas, formando uma população  $Z$  de tamanho  $2P$ .

O primeiro passo da atualização da população é a classificação da mesma em níveis de dominância, sendo o mesmo procedimento presente no NSGA-II. O primeiro nível de dominância  $F_1$  corresponde à imagem pela aplicação  $F(\mathbf{x})$  dos elementos não dominados  $Z_1$  da população  $Z$ . Para a composição do segundo nível de dominância  $F_2$ , os elementos de  $Z_1$  são retirados da população  $Z$  e o segundo nível de dominância  $F_2$  é formado pelos elementos não dominados da nova população  $Z \setminus Z_1$ . De maneira geral, o  $i$ -ésimo nível de dominância  $F_i$  é formado pelos elementos não dominados da população formada após a exclusão dos elementos dos níveis de dominância anteriores ( $Z \setminus \{Z_1 \cup \dots \cup Z_{i-1}\}$ ). Este procedimento se repete até que esgotem os elementos da população  $Z$ . Note que por este procedimento  $k$  níveis de dominância são criados e estes determinam uma partição da população  $Z$ , ou seja  $Z = \bigcup_{i=1}^k Z_i$  e  $Z_i \cap Z_j = \emptyset$ , para  $1 \leq i, j \leq k$ . A Figura 6 mostra dois níveis de dominância  $F_1$  e  $F_2$  no espaço dos objetivos e seus correspondentes  $Z_1$  e  $Z_2$  no espaço de decisão

Em seguida uma nova população  $S_i$  é criada pela união dos elementos que formam

os níveis de dominância  $F_1, \dots, F_i$  enquanto o número de elementos de  $S_i$  não exceda  $P$ . Se  $S_j = Z_1 \cup Z_2 \cup \dots \cup Z_j$  e  $S_j$  tiver  $p$  elementos, o procedimento de atualização da população se encerra e o algoritmo passa para a etapa seguinte, fazendo  $X = S_j$ . Caso contrário, se  $|S_j| < P$  e  $|S_{j+1}| > P$ , então  $P - |S_j|$  elementos de  $Z_{j+1}$  devem ser selecionados a fim de formar a população  $X$  utilizada na iteração seguinte, composta pelos elementos de  $S_j$  e os elementos escolhidos de  $S_{j+1}$ .

Caso seja necessário, a escolha dos  $P - |S_j|$  elementos de  $Z_{j+1}$  é realizada da seguinte maneira: Inicialmente os elementos de  $S_{j+1}$  são normalizados de modo que  $0 \leq f_i(\mathbf{x}) \leq 1$  e associados aos vetores do conjunto de direções de busca  $\mathbf{w}$ . Cada ponto de  $S_{j+1}$  é associado a um vetor do conjunto de direções de busca  $\mathbf{w}$  considerando a menor distância entre eles. É criado um contador de nicho  $\rho_i$ , que contabiliza o número de associações realizadas em cada vetor  $\mathbf{w}^i \in \mathbf{w}$ . Note que mais de um vetor em  $S_{j+1}$  pode estar associado a um único vetor de  $\mathbf{w}$ . Vamos denotar esta associação por  $\gamma_i = \{s_{(i,1)}, \dots, s_{(i,r)}\}$ , que indica que  $r$  vetores  $\{s_{(i,1)}, \dots, s_{(i,r)}\} \in S_{j+1}$  estão associados ao vetor  $\mathbf{w}_i \in \mathbf{w}$ , de modo que  $\rho_i = r$ .

Seja  $\Gamma$  o conjunto formado pelas associações  $\gamma_i$  que possuem o índice  $\rho_i > 0$ . Escolha  $n - |S_j|$  elementos de  $F_{j+1}$  da seguinte forma:

1. Ordene as associações  $\gamma \in \Gamma$  por seu valor de  $\rho$ , e considere a(s) de menor valor de  $\rho$ . Se para o menor valor de  $\rho$  houver mais de uma associação  $\gamma$ , escolha uma destas de forma aleatória.
2. Para a associação  $\gamma_i = \{s_{(i,1)}, \dots, s_{(i,r)}\}$  selecionada anteriormente, escolha aleatoriamente um dos vetores  $s_{(i,1)}, \dots, s_{(i,r)}$ . Retire este vetor de  $\gamma_i$ , acrescentando-o na nova população e atualize seu contador de associações :  $\rho_i = \rho_i + 1$ .
3. Repita as instruções 1. e 2. até que  $P - |S_j|$  vetores sejam escolhidos.

Note que a associação  $\gamma_i$  escolhida na instrução 2. pode ser novamente selecionada. A contagem das associações entre vetores de  $S_{j+1}$  e  $W$  é chamada de operador de preservação de nicho (*Niche-preservation Operation*) e este procedimento de seleção faz com que se escolham elementos do último nível de dominância em direções de busca pouco exploradas, evitando o acúmulo da população em alguma região do espaço de busca.

A estrutura básica do NSGA-III é apresentada no Algoritmo 3.

## 2.3 Considerações finais

A Otimização Multiobjetivo é uma importante ferramenta que orienta o tomador de decisão que se vê diante de situações com a presença de objetivos conflitantes. Se o problema apresenta uma formulação não trivial, caracterizada por funções objetivo multimodais ou do tipo caixa preta, os Algoritmos Evolutivos, em particular o Algoritmo

**Algoritmo 3:** NSGA-III

---

```

1 begin
2    $t \leftarrow 0$ ;
3    $X_t \leftarrow$  População inicial;
4    $\mathbf{w} \leftarrow$  Direções de busca;
5    $f_t(X_t) \leftarrow$  Avaliação da aptidão;
6   while critério de parada = falso do
7      $Y_t \leftarrow$  Operadores genéticos ( $X_t$ );
8      $f_t(Y_t) \leftarrow$  Avaliação da aptidão;
9      $Z_t = X_t \cup Y_t$ ;
10     $(F_1, \dots, F_k) \leftarrow$  Non Dominated Sorting ( $Z_t$ );
11     $S_t = \emptyset$ ;
12     $i = 1$ ;
13    while  $|S_t| \leq n$  do
14       $S_t = S_t \cup F_i$ ;
15       $i = i + 1$ ;
16    end while
17    if  $|S_t| = n$  then
18       $X_t = S_t$ ;
19       $t = t + 1$ ;
20    else
21       $\Gamma_t \leftarrow$  Niche-preservation Operation ( $S_t, F_i$ );
22       $X_t \leftarrow$  Atualização da população ( $S_t, F_i, \Gamma$ );
23       $t \leftarrow t + 1$ ;
24    end if
25  end while
26 end

```

---

Genético, são uma forma eficiente de se obter um conjunto de soluções suficientemente próximas da Fronteira Pareto do problema multiobjetivo. Além da convergência, uma outra característica que estes algoritmos apresentam é a diversidade da solução obtida, caracterizada pela dispersão das soluções ao longo da Fronteira Pareto.

Os algoritmos apresentados nesta seção representam as técnicas cuja eficiência é consagrada pela literatura até o presente momento. Estes algoritmos utilizam duas técnicas que se mostraram muito eficazes em problemas com mais de um objetivo: decomposição e a agregação. Apesar de serem técnicas antagônicas à primeira vista, estas metodologias se complementam e tratam de questões distintas. Enquanto a decomposição divide o problema principal em subproblemas, cada um deles se especializando em procurar soluções ótimas em uma região do espaço dos objetivos, a agregação procura reduzir a dimensionalidade dos fatores que levam à convergência das soluções candidatas. Estas técnicas são uma alternativa aos métodos que utilizavam unicamente a classificação da população em conjuntos não dominados como motor do processo de convergência.

O próximo capítulo trata de outra questão fundamental para o tomador de decisão:

a solução encontrada é robusta?

### 3 Otimização Robusta

Variações ou fatores de ruído são frequentemente classificados como ruído interno (gerado pelo sistema em estudo, seus componentes e parâmetros) e ruído externo (gerado pelo ambiente). Um exemplo de ruído interno é a fadiga dos materiais devido ao uso enquanto que o desgaste característico provocado por diferentes usuários ou ambientes é um exemplo de ruído externo.

Incertezas têm sua origem em erros humanos, efeitos físicos desconhecidos, eventos extremos não previstos, variações aleatórias no uso ou nas características do produto [12].

O caráter aleatório e imprevisível do ruído e incertezas tornam muito difícil sua eliminação e um ambiente desprovido destas características é muito restritivo ou oneroso. Desta maneira, a confiabilidade de um sistema aumenta quando este apresenta a resposta esperada mesmo com a presença de variações, ruídos e incertezas.

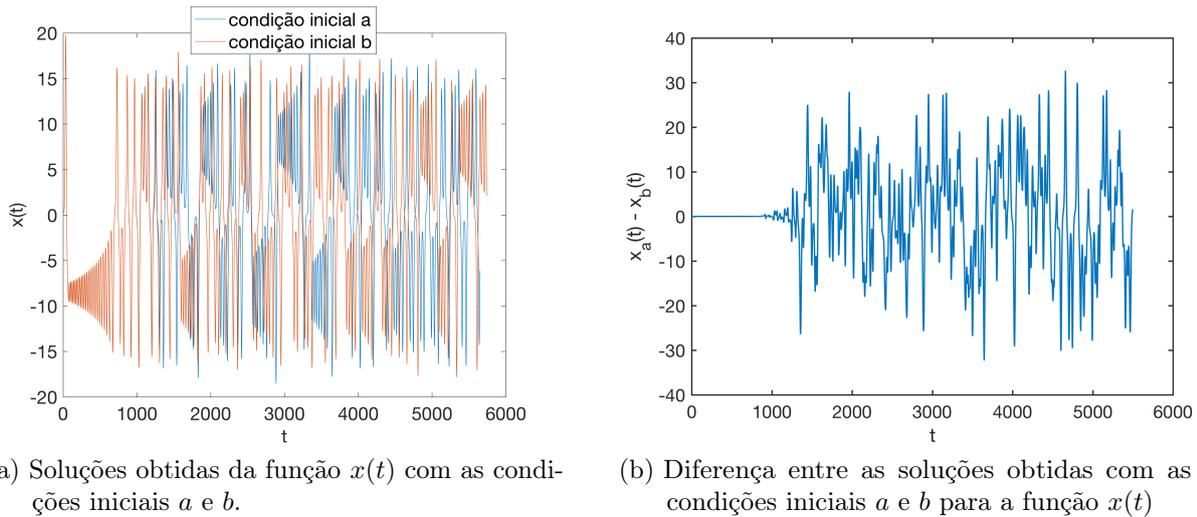
A imprevisibilidade e o ruído muitas vezes provocam resultados não desejados em um problema. Um exemplo do efeito catastrófico causado pela sensibilidade nos parâmetros de um problema pode ser observado nas equações de Lorenz (3.1), que representam um modelo simplificado para o fenômeno de convecção atmosférica [91].

$$\begin{aligned}\frac{dx}{dt} &= \sigma(y - x) \\ \frac{dy}{dt} &= x(\rho - z) - y \\ \frac{dz}{dt} &= xy - \beta z\end{aligned}\tag{3.1}$$

Neste sistema,  $x(t)$  é proporcional à intensidade do movimento convectivo,  $y(t)$  é proporcional à variação da temperatura horizontal e  $z(t)$  é proporcional à variação da temperatura vertical, enquanto que  $\sigma$ ,  $\rho$  e  $\beta$  são parâmetros do sistema.

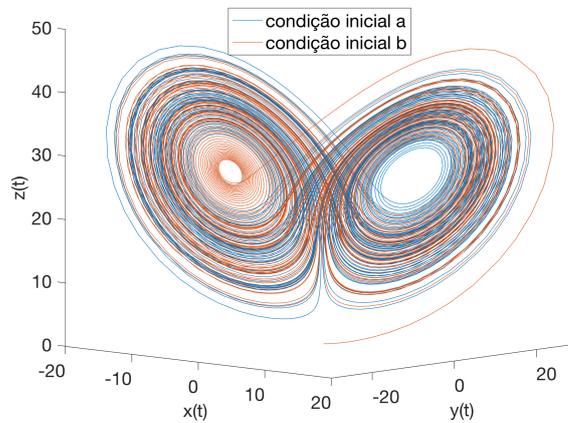
A figura 7a apresenta as soluções obtidas para a função  $x(t)$  enquanto que a figura 7c apresenta as trajetórias obtidas para as funções  $x(t)$ ,  $y(t)$  e  $z(t)$  com as condições iniciais  $a : x(0) = y(0) = z(0) = 0.2$  e  $b : x(0) = y(0) = z(0) = 0.200001$ , com  $\sigma = 10$ ,  $\beta = 8/3$  e  $\rho = 20$ . A figura 7b apresenta a diferença entre as soluções obtidas para a função  $x(t)$  com as condições iniciais  $a$  (denotado por  $x_a(t)$ ) e  $b$  (denotado por  $x_b(t)$ ). Observe que para  $0 < t < 1000$  as soluções são muito próximas. Entretanto para  $t > 1000$  a diferença entre as soluções é significativa, sendo muitas vezes maior do que o intervalo de variação individual de cada uma das funções  $x_a(t)$  e  $x_b(t)$ .

O estudo da incerteza em um sistema é conhecido como robustez [117, 13] e está



(a) Soluções obtidas da função  $x(t)$  com as condições iniciais  $a$  e  $b$ .

(b) Diferença entre as soluções obtidas com as condições iniciais  $a$  e  $b$  para a função  $x(t)$



(c) Trajetórias do atrator de Lorenz com as condições iniciais  $a$  e  $b$ .

Figura 7 – Soluções do sistema de equações (3.1) com as condições iniciais  $a$  e  $b$ .

presente em várias áreas de pesquisa. Na teoria do controle, o controle robusto visa à garantia de estabilidade dos sistemas por meio do problema de otimização das normas  $\mathcal{H}_2$  e  $\mathcal{H}_\infty$  [11, 117, 52]. Em Zang et al. [164], os parâmetros de um sistema são classificados da seguinte forma:

**fatores de sinal:** (*signal factors*) determinam as configurações a serem consideradas no controle robusto;

**fatores de ruído:** (*noise factors*) são fatores de difícil controle, causadores de variabilidade no sistema;

**fatores de controle** (*control factors*) são fatores que devem ser otimizados de modo a reduzir a sensibilidade da resposta dos fatores de ruído.

Esta composição é ilustrada no diagrama de parâmetros apresentado na figura 8.

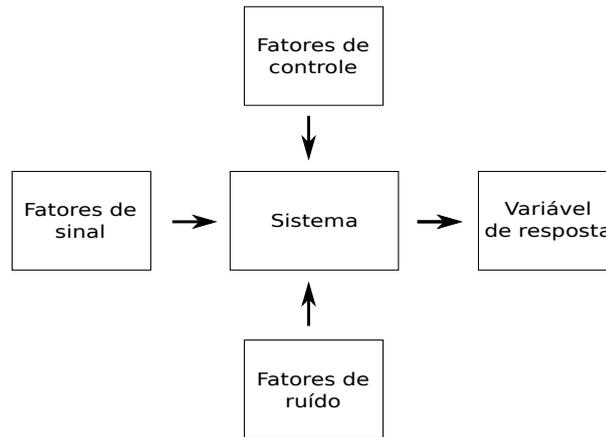


Figura 8 – Diagrama de parâmetros (*p-diagram*) [41]

O *design* Robusto [138, 139] é um conceito apresentado pelo engenheiro Genichi Taguchi que visa reduzir as variações de um produto sem eliminar as causas da variação. Sua metodologia é composta por três elementos [13]:

**System design:** determina a performance básica dos parâmetros de um produto e sua estrutura geral;

**Parameter design:** Parâmetros do problema que determinam os níveis de qualidade;

**Tolerance design:** Ajuste fino do *parameter design*.

Como o propósito do *design* robusto é a mitigação do efeito do ruído, dois parâmetros são considerados no processo de otimização: os parâmetros de controle  $\mathbf{x}$ , que são alvo do processo de otimização e os fatores de ruído  $\xi$  (condições ambientais e características dos produtos e materiais) que são fatores difíceis de serem controlados. Se  $y_i = y(\mathbf{x}, \xi_i)$  é um indicador de qualidade (mantendo  $\mathbf{x}$  constante) para o ruído  $\xi_i$  e  $\hat{y}$  é o padrão de qualidade desejado, o desvio quadrado médio (MDS) é dado pela expressão  $MDS = \frac{1}{k} \sum_{i=1}^k (y(\mathbf{x}, \xi_i) - \hat{y})^2$ . Por fim é definida a função *signal-to-noise ratio* (SNR)

$$SNR = -10 \log_{10}(MDS) \quad (3.2)$$

que deve ser maximizada em  $\mathbf{x}$ .

Uma completa lista de abordagens e métodos para a otimização robusta pode ser encontrada nos trabalhos de Ide and Schöbel [61], Kuroiwa and Lee [78], Beyer and Sendhoff [13], Gorissen et al. [48]

### 3.1 Incertezas no processo de Otimização

Esta seção apresenta a classificação das incertezas presentes em um processo de otimização. Supõe-se presente uma função objetivo  $F(\mathbf{x})$ <sup>1</sup> que modele o processo em estudo, dependente de um conjunto de variáveis representado por um vetor  $\mathbf{x}$  contido em um espaço  $\mathcal{S}$ . As categorizações apresentadas independem da natureza topológica da função objetivo  $F(\mathbf{x})$  bem como de seu conjunto de  $N$  variáveis  $X$ . Entendendo-se por espaço  $\mathcal{S}$  simplesmente o conjunto onde estas variáveis fazem sentido para o problema. Todo o entendimento a apresentado nesta seção diz respeito

Uma categorização de incertezas em problemas de otimização é apresentada por Jin and Branke [68], classificando as incertezas presentes em problemas de otimização de acordo com a maneira com que estas afetam o processo de otimização. As incertezas destacadas por estes autores se dividem em:

**Incerteza na função objetivo:** Neste caso para um mesmo vetor de parâmetros  $\mathbf{x}$  a função objetivo  $F(\mathbf{x})$  apresenta valores distintos. Esta situação está associada a erros de sensores, variações nos parâmetros que determinam o objetivo  $F(\mathbf{x})$  ou em simulações aleatorizadas. Este ruído pode ser descrito matematicamente como

$$R(\mathbf{x}, \delta) = \int_{-\infty}^{\infty} [F(\mathbf{x}) + \delta] p(\delta) d\delta, \quad \delta \sim N(0, \sigma^2) \quad (3.3)$$

onde  $\mathbf{x}$  é o vetor de parâmetros (no espaço das variáveis do problema),  $F(\mathbf{x})$  é a função de aptidão do problema,  $\delta$  é o ruído aleatório com distribuição normal de média zero e variância  $\sigma^2$ . Na prática, a equação (3.3) pode ser aproximada por

$$\hat{R}(\mathbf{x}, \delta) = \frac{1}{k} \sum_{i=1}^k [F(\mathbf{x}) + \delta^i] \quad (3.4)$$

onde  $k$  é o tamanho da amostra,  $\delta^i = (\delta_1^i, \dots, \delta_M^i)$  é um vetor de ruído com  $\delta_j^i \in N(0, \sigma^2)$  e  $\hat{R}(\mathbf{x}, \delta)$  é uma estimativa de  $R(\mathbf{x}, \delta)$ .

**Incerteza nas variáveis da função objetivo:** Neste cenário, se considera o impacto provocado no objetivo  $F(\mathbf{x})$  pela adição de ruído na solução obtida. Procuram-se soluções que sejam pouco influenciadas por perturbações em suas variáveis. O processo de busca considera uma distribuição  $p(\delta)$  onde  $\delta$  é uma variável aleatória independente com distribuição normal. Considera-se a função de aptidão dada por

$$R(\mathbf{x}, \delta) = \int_{-\infty}^{\infty} F(\mathbf{x} + \delta) p(\delta) d\delta \quad (3.5)$$

<sup>1</sup> A literatura coloca função de aptidão, de avaliação e de *fitness* como sinônimos de função objetivo. Estes termos serão utilizados desta forma neste trabalho a fim de se manter fiel aos textos originais consultados.

Uma vez que a equação (3.5) pode ser de difícil avaliação ou mesmo não ter expressão analítica conhecida, esta pode ser aproximada pela expressão

$$\widehat{R}(\mathbf{x}, \delta) = \frac{1}{k} \sum_{i=1}^k F(\mathbf{x} + \delta^i) \quad (3.6)$$

Note que esta equação se assemelha à equação (3.4). Enquanto que em 3.4 a perturbação  $\delta^i$  é adicionada à variável  $\mathbf{x}$ , em (3.6) o vetor de perturbação  $\delta^i$  é adicionado ao vetor  $\mathbf{x}$ .

**Função objetivo com variação temporal:** Também conhecido como problemas dinâmicos, com a presença de uma variação temporal dos valores da função objetivo. A função de avaliação é determinística, mas depende do tempo, ou seja

$$R(\mathbf{x}, t) = F_t(\mathbf{x}) \quad (3.7)$$

**Aproximação da função objetivo:** Nos casos onde a função a ser otimizada tem um alto custo de avaliação ou não é conhecida sua expressão analítica. Também se aplica quando a função de aptidão é obtida em modelos aproximados ou obtida por meio de experimentos ou simulações. Nestas situações, a função de aptidão considerada é dada por

$$R(\mathbf{x}) = \begin{cases} F(\mathbf{x}) & \text{se a função de aptidão original é utilizada.} \\ F(\mathbf{x}) + E(\mathbf{x}) & \text{se o meta modelo for utilizado.} \end{cases} \quad (3.8)$$

Onde  $E(\mathbf{x})$  é o erro de aproximação, de natureza determinística. Ao contrário dos dois casos anteriores, este não diminui com o aumento no tamanho da amostra pois este erro é característico do modelo e não uma variação amostral.

De forma complementar à classificação de funções de aptidão com variação temporal, Talbi [140] cita a Otimização Multi-periódica como uma classe de otimização dinâmica onde as variáveis de entrada se alteram periodicamente e esta variação é conhecida *a priori*.

Em Ong et al. [114], os autores classificam as incertezas em três categorias, considerando qual elemento do modelo é afetado: função objetivo, variáveis ou ambiente:

**Categoria I:** Nesta categoria a incerteza é resultado de ruído na função objetivo. A natureza deste ruído se deve à falta de acurácia do modelo utilizado tais como a discretização de situações de natureza contínua, utilização de meta modelos, erros e imprecisão de sensores. Assim, dado o vetor de variáveis  $\mathbf{x} \in \mathcal{S}$  e a função objetivo  $F(\mathbf{x})$ , a formulação robusta  $F'(\mathbf{x}, \delta)$  de categoria I tem a forma

$$F'(\mathbf{x}, \delta) = F(\mathbf{x}) + \delta \quad (3.9)$$

onde  $\delta$  é o parâmetro de ruído adicionado a função objetivo  $F(\mathbf{x})$ .

**Categoria II:** Nesta categoria, a incerteza está associada às variáveis  $\mathbf{x} \in \mathcal{S}$ . Este tipo de incerteza está associada por exemplo a flutuações na composição das matérias primas de um produto e incertezas em medidas ou parâmetros. Dado o vetor de variáveis  $\mathbf{x} \in \mathcal{S}$  e a função objetivo  $F(\mathbf{x})$ , a formulação robusta  $F'(\mathbf{x}, \delta)$  de categoria II tem a forma

$$F'(\mathbf{x}, \delta) = F(\mathbf{x} + \delta) \quad (3.10)$$

onde  $\delta = (\delta_1, \dots, \delta_N)$  é o vetor de ruído gaussiano adicionado à variável  $\mathbf{x} \in \mathcal{S}$ , sendo  $N$  o número de variáveis do problema.

**Categoria III:** Nesta categoria estão agrupadas as flutuações nas condições operacionais do problema que não estão associadas à função objetivo ou às variáveis. Flutuações da temperatura, umidade, desleixo de usuários finais são exemplos deste tipo de incertezas. Desta forma, dado o vetor de variáveis  $\mathbf{x} \in \mathcal{S}$  e a função objetivo  $F(\mathbf{x})$ , a formulação robusta  $F'(\mathbf{x}, \delta)$  de categoria III tem a forma

$$F'(\mathbf{x}, \mathbf{c}, \delta) = F(\mathbf{x}, \mathbf{c} + \delta) \quad (3.11)$$

onde  $\mathbf{c} = (c_1, \dots, c_N)$  é o vetor das variáveis do ambiente e  $\delta = (\delta_1, \dots, \delta_N)$  representa a variabilidade do mesmo.

Uma discussão mais detalhada sobre a natureza e a formulação das categorias II e III podem ser encontradas no trabalho de Jin and Sendhoff [69].

Além classificação das incertezas presentes em modelos temos a classificação das soluções obtidas por técnicas de Otimização Robusta. Deb and Gupta [30] apresentam duas classificações para solução de problemas de otimização robusta. Considere o problema de otimização mono-objetivo dado por

$$\begin{aligned} &\text{Minimize } f(\mathbf{x}) \\ &\text{sujeito a } \mathbf{x} \in \mathcal{S} \end{aligned} \quad (3.12)$$

Onde  $\mathcal{S}$  é o espaço de busca factível. Para Deb, uma solução robusta é aquela que é pouco sensível a perturbações provocadas em uma vizinhança de um ponto ótimo no espaço de busca. A figura 9 apresenta uma solução global (ponto B) e uma solução robusta (ponto A) para o problema de minimização mono-objetivo da função  $F(\mathbf{x})$ . Note que a função apresenta uma variação menor na solução robusta quando comparada à solução global, caracterizando a menor sensibilidade da função à variação do seu parâmetro.

Deb and Gupta [30] classificam a robustez da solução de um problema de otimização é classificada de acordo com sua formulação. Para o problema de otimização mono-objetivo

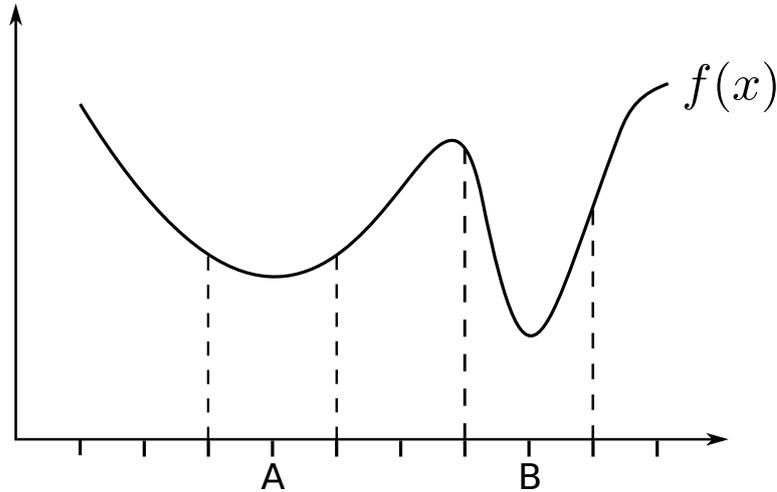


Figura 9 – Solução robusta (A) e solução global (B) em um problema de minimização mono-objetivo

onde se deseja a minimização de uma função  $F(\mathbf{x})$  com  $\mathbf{x} \in S$ , uma solução robusta será do tipo I ou do tipo II:

**Tipo I:** Uma solução será do Tipo I se for obtida pela minimização do valor médio de  $f(\mathbf{x})$  em uma vizinhança  $B_\delta(\mathbf{x})$  definida, ou seja, considera-se o problema de minimização da função média efetiva dado por

$$\begin{aligned} \text{minimize } f^{eff}(\mathbf{x}) &= \frac{1}{|B_\delta(\mathbf{x})|} \int_{\mathbf{y} \in B_\delta(\mathbf{x})} f(\mathbf{y}) d\mathbf{y} & (3.13) \\ \text{sujeito a } &\mathbf{x} \in S \end{aligned}$$

sendo  $|B_\delta(\mathbf{x})|$  o hiper-volume da vizinhança  $B_\delta$  de  $\mathbf{x}$ .

**Tipo II:** Soluções robustas do Tipo II são aquelas obtidas por problemas de minimização onde o tratamento das perturbações ocorre como uma restrição do problema de minimização, ou seja, consideramos o problema de minimização dado por

$$\begin{aligned} \text{minimize } &f(\mathbf{x}) & (3.14) \\ \text{sujeito a } &\frac{|f^p(\mathbf{x}) - f(\mathbf{x})|}{|f(\mathbf{x})|} \leq \eta, \mathbf{x} \in S \end{aligned}$$

onde a função perturbada  $f^p(\mathbf{x})$  pode ser a função média efetiva  $f^{eff}(\mathbf{x})$  ou o pior caso observado em uma dada vizinhança  $B_\delta(\mathbf{x})$ , dentre outras.

Neste mesmo artigo, os autores estendem esta formulação para problemas multiobjetivo e apresentam um conjunto de funções de *benchmark* para Otimização Multiobjetivo Robusta.

Em Goh et al. [46] os autores consideram o problema de otimização mono-objetivo (*Single Objective Optimization - SO*), classificando as metodologias de obtenção de soluções de problemas robustos (e conseqüentemente, classificando suas soluções) de duas maneiras:

**SO approach:** O processo de otimização é realizado por meio das medidas de robustez substituindo o objetivo original. Ainda temos um problema de otimização mono-objetivo, mas este utiliza alguma medida de robustez  $F(x)$  (apresentadas na seção 3.2) no lugar da função objetivo original  $f(x)$ , sendo portanto uma extensão da abordagem proposta por Deb & Gupta em sua formulação de solução do Tipo I.

**MO approach:** O processo de otimização é realizado por meio da adição das medidas de robustez como objetivos adicionais, tornando o problema de otimização mono-objetivo em um problema de otimização multiobjetivo (*Multi Objective Optimization - MO*). Neste caso não se deve fazer qualquer analogia com a formulação do Tipo II proposta por Deb & Gupta, que usa uma medida de robustez é considerada como uma restrição e não como um problema adicional. Esta abordagem é utilizada por Guimarães et al. [51]

## 3.2 Medidas de robustez

A equação (3.6) (e por similaridade a equação (3.4)) é apontada por Goh et al. [46] como uma expectativa de medida de performance em problemas robustos. Estas são expressões também conhecidas como de Integração de Monte Carlo [156], que por se tratar de um processo amostral eleva o número de avaliações da função objetivo na vizinhança do ponto considerado. Por outro lado, a estimativa de erro reduz com o aumento do tamanho da amostra <sup>2</sup>.

Outras medidas de robustez são apresentadas na literatura. Jin and Sendhoff [69] afirmam que para um problema mono-objetivo  $f(\mathbf{x})$  estas se dividem em duas categorias:

1. Medidas de valor médio na avaliação do objetivo em uma vizinhança, dada por

$$R(\mathbf{x}, \mathbf{w}, \delta) = \frac{\sum_{i=1}^k w_i f_i(\mathbf{x} + \delta_i)}{\sum_{i=1}^k w_i} \quad (3.15)$$

onde  $\delta_i$  corresponde a um vetor de pequenas perturbações estocásticas e  $w_i$  um vetor de pesos, podendo ser  $w_i = 1$ ,  $1 \leq i \leq k$ .

<sup>2</sup> O erro amostral é dado por  $\frac{\sigma^2(F)}{k}$  sendo  $k$  o tamanho da amostra e  $\sigma^2(F)$  a variância da função, definida como  $\sigma^2(F) = \int (F(\mathbf{x}) - I)^2 d\mathbf{x}$  onde  $I = \int F(\mathbf{x}) d\mathbf{x}$ .

2. Medidas de perturbação, que é um caso particular da medida de variação apresentada acima, onde se desconsidera o vetor de pesos  $w_i$  e se associa uma única perturbação por ponto:

$$R(\mathbf{x}, \delta) = f(\mathbf{x} + \delta(t)) \quad (3.16)$$

onde  $t$  representa a iteração <sup>3</sup>. O vetor de perturbação  $\delta(t)$  deve ser gerado a cada avaliação.

Outras medidas de valor médio são citadas por Gaspar-Cunha and Covas [42], Ferreira et al. [36], utilizando o valor médio da derivada discreta em uma vizinhança de um ponto  $\mathbf{x}_i$ . Esta medida de robustez, que mede a variância da função, é dada por

$$R(\mathbf{x}_i) = \frac{1}{k} \sum_{j=0}^k \left| \frac{\tilde{f}(\mathbf{x}_j) - \tilde{f}(\mathbf{x}_i)}{\mathbf{x}_j - \mathbf{x}_i} \right|, \quad d_{i,j} < d_{\max} \quad (3.17)$$

onde  $\tilde{f}(\mathbf{x}_i)$  é a função de aptidão normalizada dada por  $\tilde{f}(\mathbf{x}_i) = \frac{f(\mathbf{x}_i) - f_{\min}}{f_{\max} - f_{\min}}$  para problemas de maximização e  $\tilde{f}(\mathbf{x}_i) = 1 - \frac{f(\mathbf{x}_i) - f_{\min}}{f_{\max} - f_{\min}}$  em problemas de minimização,  $f_{\min}$  e  $f_{\max}$  correspondem aos limites de variação da função objetivo  $f(\mathbf{x})$  e  $k$  o número de pontos cuja distância euclidiana é menor do que um valor de referência  $d_{\max}$ , ou seja  $d_{i,j} < d_{\max}$ .

Uma outra medida de robustez é a minimização do pior caso (*worst case minimization*), ou problema minimax. A minimização do pior caso consiste no problema de otimização minimax dado por

$$R(\mathbf{x}, \delta) = \max_{\delta \in \Delta} f(\mathbf{x}, \delta) \quad (3.18)$$

sendo  $f(\mathbf{x}, \delta)$  a função objetivo,  $\mathbf{x}$  o vetor de variáveis do problema,  $\delta$  o vetor de incertezas [168, 21, 114].

As medidas de robustez apresentadas são ilustradas nas figuras 10, 11 e 12. Foi considerada uma função de avaliação hipotética (adaptada de Goh et al. [46]) que modela um problema de minimização. Este problema apresenta três mínimos locais no intervalo [0.1, 0.2], um mínimo global em  $\mathbf{x} = 0.5$  e um mínimo local em  $\mathbf{x} = 0.78$ . Destes, a solução que é menos sensível a variações do parâmetro  $\mathbf{x}$  é o mínimo local em  $\mathbf{x} = 0.78$ .

A figura 10 foi obtida pela aplicação da expressão (3.6) que mede o valor médio da função em uma vizinhança de um ponto  $\mathbf{x}$  e apresenta valores próximos para a medida de robustez em todos os ótimos locais e no ótimo global.

<sup>3</sup> Este índice de iteração (ou geração) se aplica a algoritmos evolutivos.

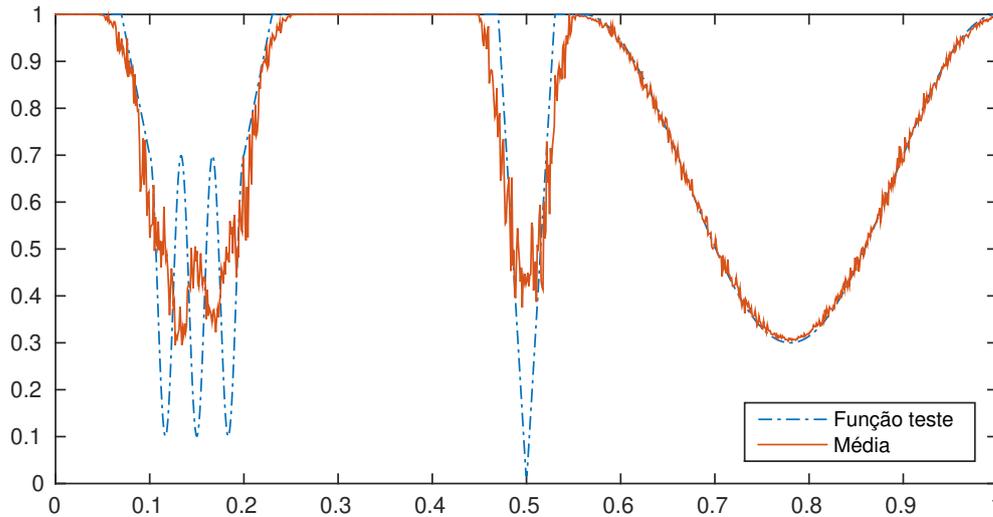


Figura 10 – Medida de robustez pelo valor médio da função.

A figura 11 utiliza a expressão (3.17) que mede a variação da função na vizinhança de um ponto  $\mathbf{x}$ . Os valores obtidos para esta medida de robustez foram normalizados no intervalo  $[0, 1]$  para melhor visualização e os menores valores correspondem à maior robustez. Note que os pontos contidos nos intervalos  $[0, 0.07]$  e  $[2.03, 0.47]$ , onde a função não apresenta variação, correspondem à máxima robustez, apesar de serem valores de máximo para esta função. No caso dos mínimos locais, o ponto de maior robustez ( $\mathbf{x} = 0.78$ ) é bem avaliado mas é indistinguível, por esta medida de robustez, dos valores localizados no intervalo  $[0.3, 0.4]$  que correspondem ao valor máximo da função e não são a resposta adequada para um problema de minimização, o que indica que esta medida de robustez deve ser utilizada de forma complementar a alguma técnica que conduza o processo de minimização.

A figura 12 ilustra a medida de robustez dada pela minimização do pior caso em uma vizinhança de um ponto  $\mathbf{x}$ , onde a solução mais robusta apresenta um *score* diferenciado dos demais pontos de mínimo locais/global.

Para todas as figuras foi considerado um vetor de perturbação com 30 valores  $\delta_j$  amostrados segundo uma distribuição uniforme no intervalo  $[-0.025, 0.025]$  utilizando um hipercubo latino (unidimensional) e em seguida foram utilizados os valores perturbados da forma  $\mathbf{x}_i + \delta_j$  (ou seja, temos  $k = 30$  nas fórmulas (3.6) e (3.17)). No caso da minimização do pior caso, em cada ponto  $\mathbf{x}_i$  do espaço de decisão foi considerado o maior valor da perturbação local dada por  $F(\mathbf{x}_i + \delta_j)$  com  $j = 1, \dots, 30$ .

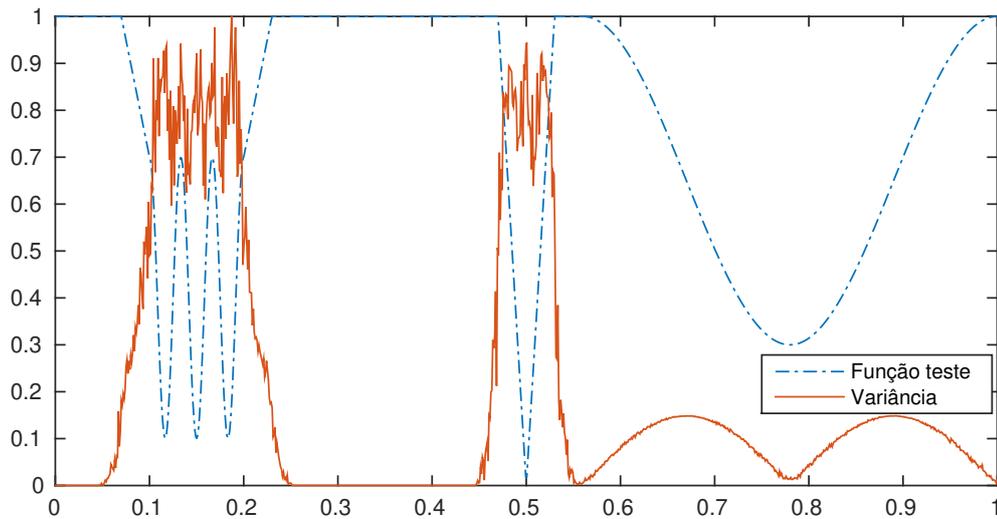


Figura 11 – Media de robustez pela variância (valores normalizados).

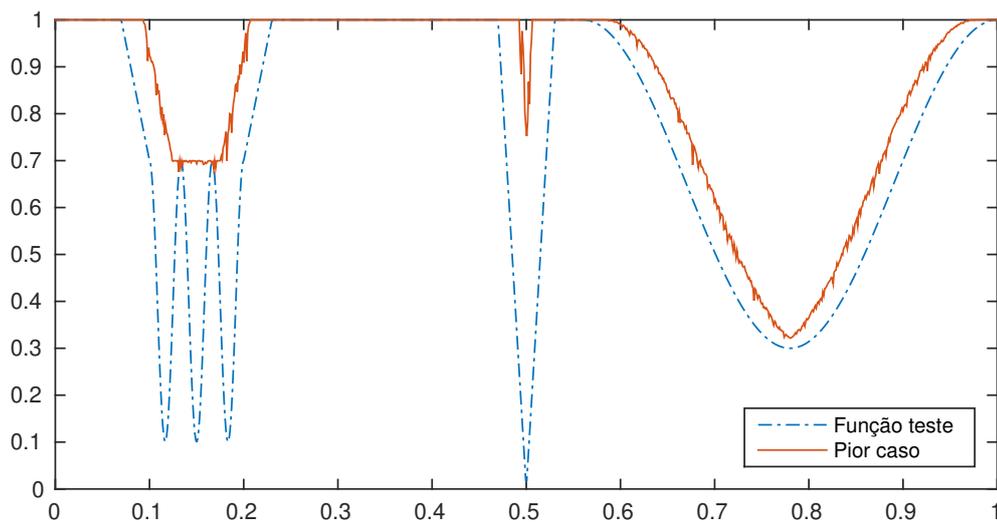


Figura 12 – Medida de robustez pelo pior caso.

### 3.3 Incorporação da robustez no processo de otimização

Em Cunha et al. [22], os autores apresentam a otimização de forma mais geral, como “o campo de conhecimentos cujas técnicas visam determinar os extremos (máximos ou mínimos) de funções, em domínios determinados”. Rakshit et al. [120] apresentam as seguintes classificações para os processos de otimização que consideram a incerteza e a variabilidade:

**Otimização ruidosa:** Ou *Noisy Optimization*, que corresponde à otimização de funções com a presença de ruído no espaço dos objetivos, ou seja, para problemas de otimização da forma  $F(\mathbf{x}, \delta) = F(\mathbf{x}) + \delta$ , sendo  $\mathbf{x}$  o vetor de parâmetros do problema

e  $\delta$  o ruído. Sua medida de robustez é dada pela equação (3.3) e de forma simplificada, pela equação (3.4)).

**Otimização Robusta:** Ou *Robust Optimization*, que corresponde à otimização de funções com a presença de ruído no espaço de decisão, ou seja, para problemas de otimização da forma  $F(\mathbf{x}, \delta) = F(\mathbf{x} + \delta)$ , sendo  $\mathbf{x}$  o vetor de parâmetros do problema e  $\delta$  o ruído. Sua medida de robustez é dada pela equação (3.5) e de forma mais simplificada pela equação (3.6)).

**Otimização Dinâmica:** Ou *Dynamic Optimization*, que corresponde à otimização de funções com variação temporal, ou seja, para problemas de otimização da forma  $F(\mathbf{x}, t) = F_t(\mathbf{x})$  onde a função objetivo varia com o tempo. No caso da otimização multi-periódica, esta mudança ocorre de forma periódica, que é conhecida *a priori*. Sua formulação matemática é dada pela equação (3.7).

**Fitness approximation:** Corresponde à aproximação de funções por meio de um meta-modelo, aplicado em situações onde a função original apresenta um elevado custo computacional. [67]. Sua formulação matemática é dada pela equação (3.8).

A maneira usual de se incorporar incertezas e variações é a utilização das medidas de robustez descritas na seção 3.2 como a função de *fitness*, criando versões robustas dos algoritmos já consagrados na literatura tais como o RNSGA-II ([30, 27]) e o RMOPSO ([106, 19]). Há três formas básicas de incorporar o ruído nas funções de *fitness*: a Média Implícita, a Média Explícita e a estimativa do pior caso:

**Média Explícita:** Dada uma função objetivo  $f(x)$ , para cada um dos  $n$  elementos da população  $x_i$ , considera-se um conjunto de  $k$  vetores de perturbação  $\delta_{i,j} = \{\delta_{i,1}, \delta_{i,2}, \dots, \delta_{i,k}\}$ . O valor estimado da função com a incorporação do ruído o ponto  $x_i$  é dado por

$$R(\mathbf{x}_i, \delta_i) = \frac{1}{k} \sum_{j=1}^k f(\mathbf{x}_i + \delta_{i,j}) \quad (3.19)$$

A escolha dos vetores de perturbação  $\delta_{i,j}$  pode estar associada a uma vizinhança  $B(\mathbf{x}_i, \epsilon)$ <sup>4</sup> de centro em  $\mathbf{x}_i$  e raio  $\epsilon$  de modo que  $\mathbf{x}_i + \delta_{i,j} \in B(\mathbf{x}_i, \epsilon)$ . Esta amostragem deve ser aleatória, sendo comum a utilização de um Hipercubo Latino, um Hipercubo Latino Ortogonal [142] ou um Hipercubo Latino Aperfeiçoado<sup>5</sup>. A figura 13 ilustra a diferença entre amostras aleatórias utilizando uma distribuição uniforme, um

<sup>4</sup> Uma vizinhança (ou bola aberta) de centro em  $\mathbf{x}$  e raio  $\epsilon$  em um espaço  $X$  é definida como  $B(\mathbf{x}, \epsilon) = \{\mathbf{y} \in X : \|\mathbf{x} - \mathbf{y}\|_p < \epsilon\}$ , sendo  $\|\cdot\|_p$  a norma  $p$ . Para  $p = 1$  temos a norma da soma, para  $p = 2$  temos a norma euclidiana usual e para  $p = \infty$  temos a norma do máximo.

<sup>5</sup> Uma implementação em MATLAB para amostragens utilizando o Hipercubo Latino (*Improved hypercube Sampling* - IHS) está disponível em [http://people.sc.fsu.edu/~jburkardt/m\\_src/ihs/ihs.html](http://people.sc.fsu.edu/~jburkardt/m_src/ihs/ihs.html)

hipercubo latino, um hipercubo latino ortogonal e um conjunto de pontos igualmente distribuídos. O efeito do ruído em amostras e seu impacto em um Algoritmo genético pode ser encontrado em [105, 120].

**Média implícita:** Consiste em simplesmente aumentar o tamanho da população, compensando o efeito do ruído em um ponto pela avaliação de diversos pontos próximos.

**Estimativa do Pior caso:** Para a minimização do pior caso, pode-se utilizar como medida de robustez  $R(\mathbf{x}_0, \delta, \epsilon)$  o maior valor da função  $f(\mathbf{x})$  em um intervalo de centro em  $\mathbf{x}_0$  e raio  $\epsilon$ :

$$R(\mathbf{x}_0, \delta, \epsilon) = \max_{\delta \in [-\epsilon, \epsilon]} f(\mathbf{x}_0 + \delta) \quad (3.20)$$

Para esta estimativa, pode-se considerar uma amostra de tamanho  $k$  com distribuição uniforme no intervalo  $[x_0 - \epsilon, x_0 + \epsilon]$ .

Gaspar-Cunha and Covas [42] propõem como alternativa à média amostral dada pela equação 3.19 a expressão

$$R(x_i) = \left( 1 - \frac{\sum_{j=0}^k |\tilde{f}(x_j) - \tilde{f}(x_i)|}{N} \right) f(x_i) \quad (3.21)$$

onde  $\tilde{f}(x_i)$  é a função de aptidão normalizada dada por  $\tilde{f}(x_i) = \frac{f(x_i) - f_{\min}}{f_{\max} - f_{\min}}$  para problemas de maximização e  $\tilde{f}(x_i) = 1 - \frac{f(x_i) - f_{\min}}{f_{\max} - f_{\min}}$  em problemas de minimização,  $f_{\min}$  e  $f_{\max}$  correspondem aos limites de variação da função objetivo  $f(\mathbf{x})$  e  $k$  o número de pontos cuja distância euclidiana é menor do que um valor de referência  $d_{\max}$ , ou seja  $d_{i,j} < d_{\max}$ .

Uma consequência da utilização de médias implícitas ou explícitas é o aumento do número de avaliações das funções objetivo, o que pode ser desastroso no caso de funções cuja avaliação tenham um custo computacional muito alto. Algumas alternativas são apresentadas como remédio para estas situações. Kapelan et al. [71] apresentam uma versão robusta do algoritmo NSGA-II (RNSGA-II) aplicada ao *design* robusto do sistema de distribuição de água utilizando uma variação da técnica da média implícita. Para reduzir o número de avaliações, é realizada uma amostragem de Monte Carlo em cada ponto do Conjunto Pareto Ótimo do problema no término do algoritmo e os novos pontos não dominados são incorporados à população. Outra técnica que visa à redução do custo computacional é a substituição da função objetivo original por um meta modelo. Zhou and Zhang [168] utilizam o Processo Gaussiano de Regressão (também conhecido como Kriging [167, 86]) para criar um metamodelo (*meta-model*)  $y = \bar{f}(x)$  visando reduzir o número

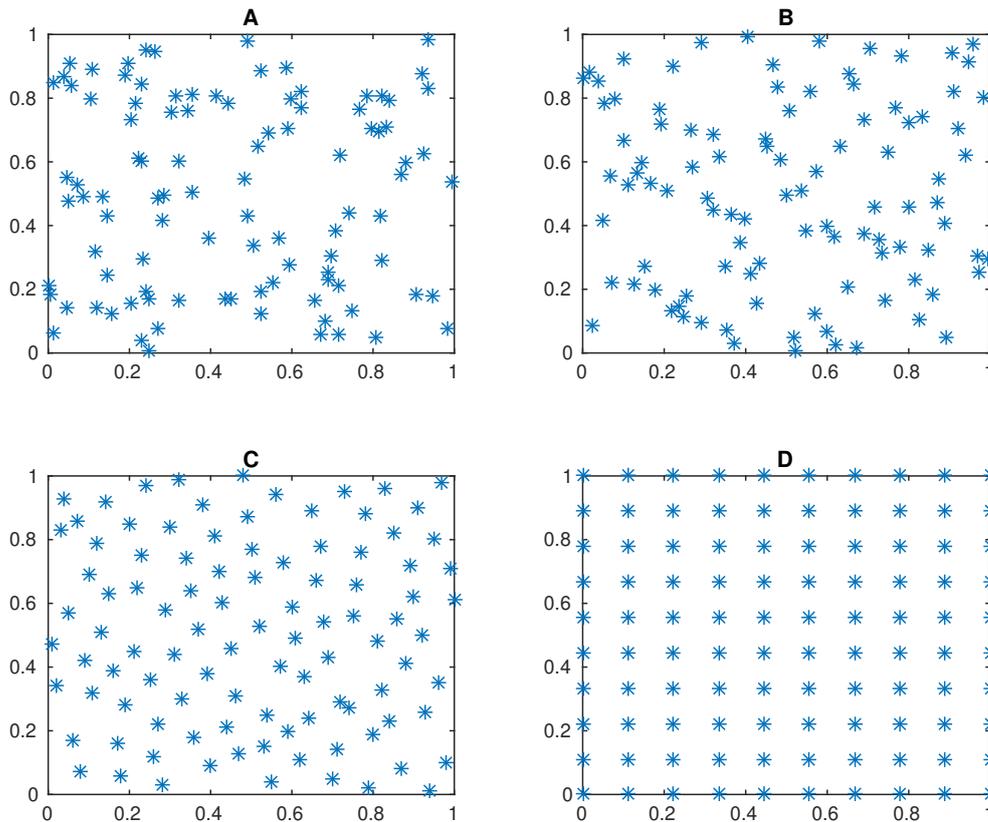


Figura 13 – Uma amostra de 100 pontos no quadrado unitário  $[0, 1] \times [0, 1]$ : A figura **A** apresenta uma amostragem utilizando uma distribuição uniforme. A figura **B** é uma amostragem utilizando o Quadrado Latino, a figura **C** é uma amostragem utilizando o Quadrado Latino Aperfeiçoado (*Improved Hypercube sampling* – [8]) e a figura **D** apresenta pontos igualmente distribuídos.

de avaliações da função objetivo original em problemas robustos do tipo **minimax**. Para o mesmo tipo de problema de otimização robusta e com o mesmo propósito Ong et al. [114] utilizam funções de base radial (*radial basis functions* - RBF) e aplicam o algoritmo proposto no *design* de aerofólio de um avião e Paenke et al. [116] utilizam regressão e interpolação polinomial quadrática para se obter uma aproximação da função de *fitness*.

Um exemplo de incorporação de um algoritmo minimax aplicado à otimização robusta é descrito em Asafuddoula et al. [4]. Neste trabalho, os autores definem uma quantificação da robustez de uma solução por meio da razão  $\sigma_g = \mu_g / \sigma_g$  para as restrições do problema, sendo  $\mu$  o valor médio em um ponto e  $\sigma$  o desvio padrão, e  $\sigma_f = \sigma_{f0} / \sigma_f$  para os objetivos, sendo  $\sigma_{f0}$  um valor definido pelo usuário (que controla a variação da função objetivo) e  $\sigma_f$  o desvio padrão da função objetivo. A média e os desvios são obtidos por meio de média explícita, e as razões  $\sigma_f$  e  $\sigma_g$  devem ser maximizados para se obter a melhor robustez. Asafuddoula et al. apresentam quatro algoritmos, que consistem na incorporação do problema minimax dado por  $\max \{ \min(\sigma_f, R_f) \}$  e/ou

$\max \{ \min(\sigma_g, R_C) \}$  como objetivos extras, sendo  $R_f$  e  $R_c$  valores limite para a variação das funções ou dos objetivos.

Além das metodologias de incorporação das medidas de robustez nas funções a serem otimizadas, como restrições ou objetivos adicionais, outras propostas são apresentadas. Li et al. [84] apresentam um algoritmo competitivo coevolutivo aplicado ao *design* robusto de um motor de ímã permanente, procurando a minimização do pior caso. Soares et al. [134] apresentam um algoritmo evolucionário multiobjetivo para otimização robusta ([I]MOEA), aplicado ao problema de minimização do pior caso, utilizando análise intervalar ([110]).

### 3.4 Otimização Robusta Multiobjetivo

Esta seção apresenta o Problema de Otimização Robusta com presença de ruído nas variáveis. Considere o Problema de Otimização Multiobjetivo dado por

$$\mathbf{x}^* = \arg \min F(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_M(\mathbf{x})) \quad (3.22)$$

$$\text{sujeito a: } \begin{cases} g_i(\mathbf{x}) \leq 0, & 1 \leq i \leq p \\ \mathbf{x} \in \mathcal{S} \end{cases} \quad (3.23)$$

sendo  $g_i(\mathbf{x})$   $p$  restrições de desigualdade. Para esta formulação, definimos o problema de otimização multiobjetivo robusto

$$\text{Minimize } R(\mathbf{x}, \delta) = (R_1(\mathbf{x}, \delta), \dots, R_M(\mathbf{x}, \delta)) \quad (3.24)$$

$$\text{sujeito a } \begin{cases} R_{g_i}(\mathbf{x}) \leq 0, & 1 \leq i \leq p \\ \mathbf{x} \in \mathcal{S} \subset \mathbb{R}^N \\ \delta \in \Delta \subset \mathbb{R}^N \end{cases} \quad (3.25)$$

sendo  $R_i(\mathbf{x}, \delta)$ ,  $1 \leq i \leq M$  uma medida de robustez para o objetivo  $f_i(\mathbf{x})$  do problema  $F(\mathbf{x})$  com  $M$  objetivos,  $R_{g_i}(\mathbf{x})$  uma medida de robustez para a restrição de desigualdade  $g_i(\mathbf{x})$ ,  $\mathcal{S}$  o conjunto das  $N$  variáveis de decisão,  $\Delta$  o conjunto das  $N$  variáveis de perturbação,  $g_i(\mathbf{x}, \delta)$ ,  $1 \leq i \leq p$  as  $p$  restrições de desigualdade. Não são consideradas restrições de igualdade, apesar deste tipo de restrição ser considerado por alguns autores [121, 162].

Uma solução do Problema de Otimização Robusta é determinada pela relação de ordem parcial de Pareto da sua medida de robustez  $R(\mathbf{x}, \delta)$  no Espaço dos Objetivos. Uma vez que as medidas de robustez são fortemente influenciadas pela amostra de pontos em uma vizinhança (e também pela própria vizinhança) de um ponto  $\mathbf{x}$ , não há garantia de otimalidade. Este conjunto de soluções forma uma Fronteira de Pareto Robusta, que pode ou não coincidir com a Fronteira Pareto usual de um problema de otimização usual com

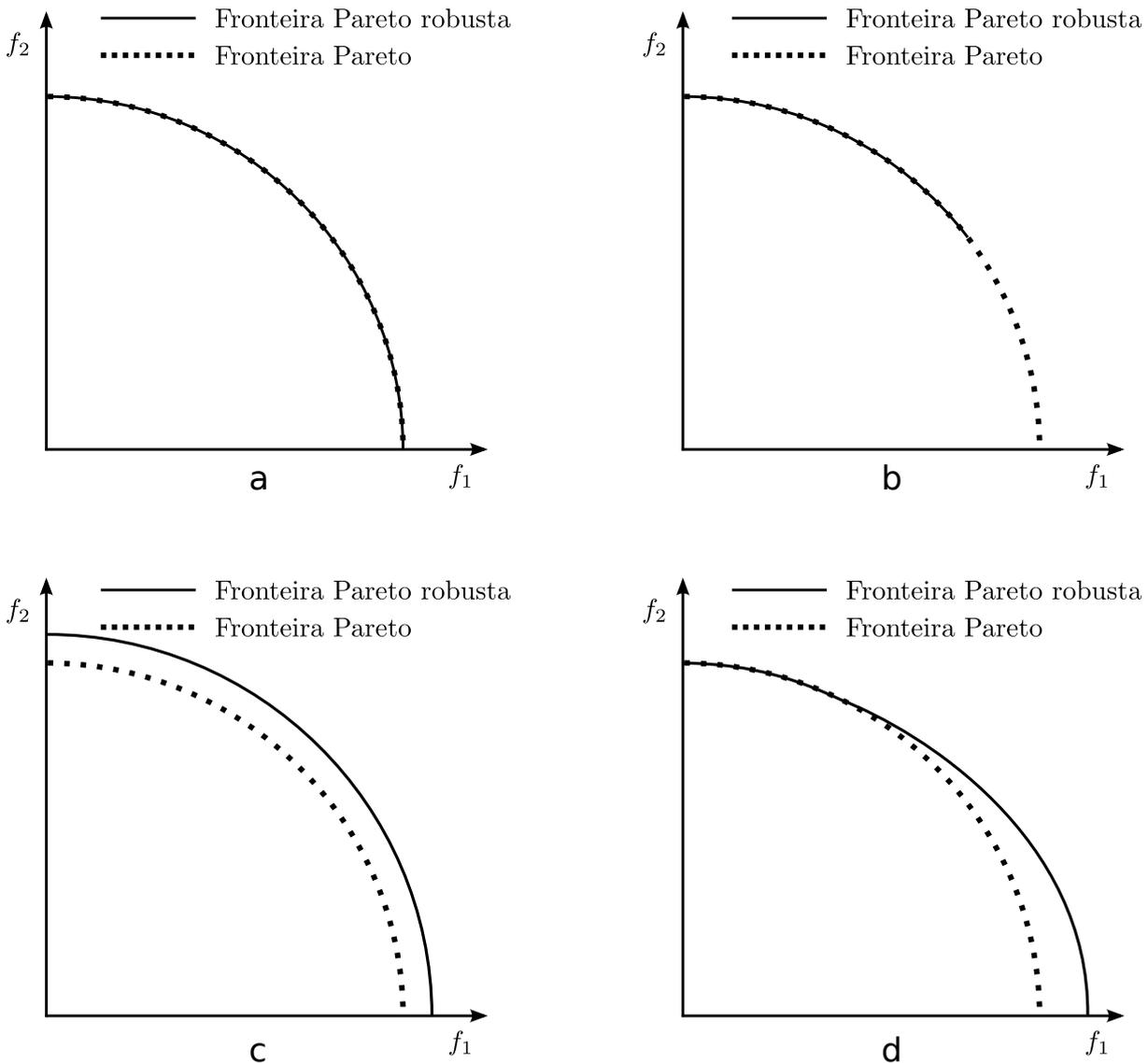


Figura 14 – Fronteira Pareto robusta e usual

as mesmas características, porém desconsiderando-se o ruído  $\delta \in \Delta$  nas variáveis  $\mathbf{x} \in \mathcal{S}$ . As Fronteiras Pareto usual e robusta podem se relacionar da seguinte forma [30, 43]:

- As Fronteiras Pareto usual e robusta são iguais;
- A Fronteira Pareto usual contém a Fronteira Pareto robusta, mas não são iguais;
- A Fronteira Pareto usual e a Fronteira Pareto robusta são disjuntas;
- Uma parte da Fronteira Pareto robusta é igual à Fronteira Pareto usual e a outra Parte é dominada pela Fronteira Pareto usual.

As posições relativas das Fronteiras Pareto usual e robusta são ilustradas na figura 14. Ao contrário da Fronteira Pareto que é unicamente determinada pelo problema de estudo, a Fronteira Pareto Robusta varia de acordo com a medida de robustez utilizada

(como pode ser observado nas figuras 10, 11 e 12). Além disto, a Fronteira Pareto Robusta é alterada quando os parâmetros de uma medida de robustez variam. Como exemplo, considere o problema mini-max dado por

$$F(\mathbf{x}, \delta) = \min_{\mathbf{x} \in X} \left\{ \max_{\delta \in \Delta} f(\mathbf{x} + \delta) \right\} \quad (3.26)$$

Sua medida de robustez  $R(\mathbf{x}, \epsilon)$  mede o pior caso de uma função  $f(\mathbf{x})$  em um intervalo de centro em  $\mathbf{x}_0$  e raio  $\epsilon$ . Nesta situação, procuramos o maior valor de  $f(\mathbf{x})$  em uma vizinhança de centro em  $\mathbf{x}_0$  e raio  $\epsilon$ , ou seja,

$$R(\mathbf{x}_0, \delta, \epsilon) = \max_{\delta \in [-\epsilon, \epsilon]} f(\mathbf{x}_0 + \delta) \quad (3.27)$$

É fácil ver que o valor de  $R(\mathbf{x}_0, \delta, \epsilon)$  varia de acordo com o diâmetro  $2\epsilon$  desta vizinhança. A figura 15 ilustra esta situação. Desta forma, o nível de robustez desejado é um fator delicado, que determina a qualidade da resposta procurada. Este nível de robustez deve ser analisado criteriosamente pelo tomador de decisão, estando intimamente relacionado com a natureza do problema e os resultados esperados.

Uma vez que a solução do problema de Otimização Robusta Multiobjetivo depende da medida de robustez  $R(\mathbf{x}, \delta)$  utilizada, bem como dos parâmetros desta medida de robustez, considere o problema de Otimização Robusta Multiobjetivo dado em 3.24, com sua robustez definida pelo problema minimax da minimização do pior caso como descrito em 3.26 quando  $\Delta$  é uma vizinhança do ponto  $\mathbf{x}$  de raio  $\epsilon$  em  $\mathbb{R}^N$ , ou seja <sup>6</sup>

$$\Delta = [-\epsilon, \epsilon] = \prod_{j=1}^N [-\epsilon_j, +\epsilon_j] \quad (3.28)$$

### 3.5 Considerações finais

A incerteza é um fator que sempre esteve presente em problemas de natureza prática e sempre é considerado por um engenheiro em todas as etapas de seus projetos. Mesmo que o tomador de decisão não incorpore os efeitos do ruído e das incertezas na etapa de **Elaboração do Modelo** ou não utilize uma técnica de otimização que incorpore o ruído ou outros fatores incertos e de natureza aleatória, as soluções escolhidas são submetidas a uma etapa de análise de sensibilidade, que procura verificar, dentre outras coisas, o impacto da incerteza e do ruído nas variáveis e nos parâmetros do problema. Os resultados desta etapa extra podem, na pior das hipóteses, inviabilizar todas as soluções obtidas na etapa de **Otimização**. Uma outra situação indesejável seria a exclusão de uma

<sup>6</sup> Matematicamente a vizinhança citada é uma bola fechada  $B(\mathbf{x}_0, \epsilon) = \{x \in \mathbb{R}^n, \|\mathbf{x} - \mathbf{x}_0\|_\infty \leq \epsilon\}$  de centro em  $\mathbf{x}_0$  e raio  $\epsilon$ , sendo  $\|\cdot\|_\infty$  a norma de Tchebycheff (ou norma infinita, ou do supremo).

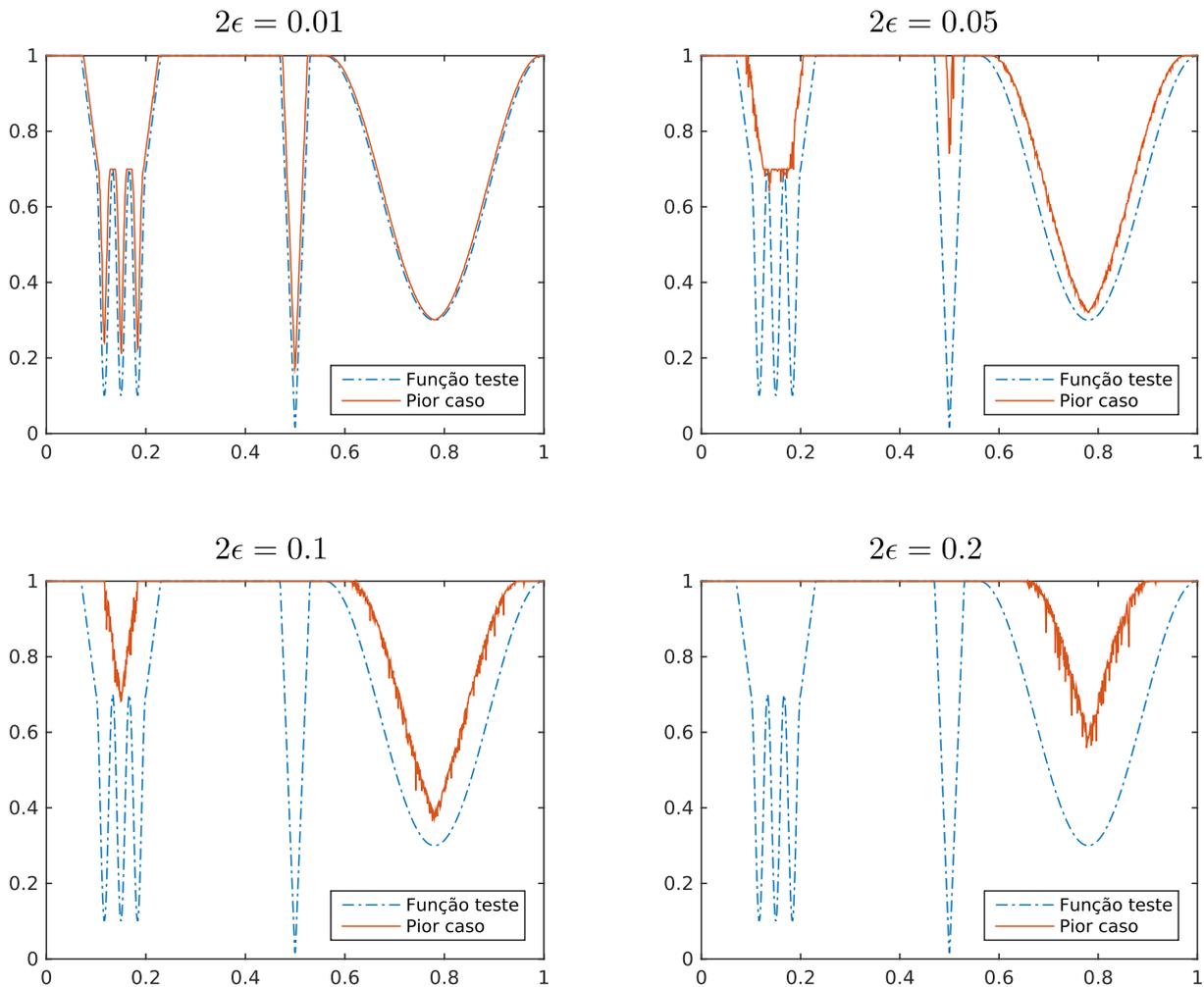


Figura 15 – Variação da medida de robustez de acordo com o diâmetro  $2\epsilon$  do intervalo indicado na equação 3.27. Foram considerados diâmetros iguais a 0.01, 0.05, 0.1 e 0.2.

solução robusta satisfatória por estar dominada por uma outra solução ótima no sentido da Dominância de Pareto, mas sensível a incertezas a ponto de ser desclassificada como opção a ser implementada.

O objetivo da Otimização Robusta é apresentar ao tomador de decisão soluções subótimas no sentido da convergência na Fronteira Pareto, que apresentem uma boa variabilidade e que sejam insensíveis às incertezas e ruídos presentes no processo. Apesar de sua importância prática, sua presença na literatura é moderada e concentrada em um único objetivo.

## 4 Um novo conjunto de funções de teste

Diversos fatores caracterizam um bom conjunto de soluções de um problema multiobjetivo. De maneira geral, a principal característica é a factibilidade. Uma solução deve, antes de tudo, ser compatível com as características e limitações do problema proposto para que esta possa ser implementada. A factibilidade é facilmente verificada: basta que as soluções apresentadas satisfaçam as restrições do problema, tanto no espaço de decisão quanto no espaço dos objetivos.

Outros dois fatores são a convergência e a dispersão, que são características inerentes ao espaço dos objetivos. Por dispersão se entende a presença de soluções ao longo de toda a Fronteira Pareto do problema e garante ao tomador de decisão a possibilidade de se analisar os diversos cenários apresentados. Por convergência se entende que as soluções obtidas devem estar o mais próximo possível da Fronteira Pareto do problema. Desta forma, graças à relação de ordem parcial dada pela dominância de Pareto, se garante que aquela solução em particular apresenta o melhor *trade off* para o cenário previamente escolhido.

Um outro fator a ser destacado é a robustez da solução encontrada. Em problemas do mundo real a presença de ruído, anomalias, variabilidade e outros fatores não previstos é a regra, e não a exceção. Tais fatores devem ser incorporados no modelo que representa o problema estudado e a solução escolhida deve minimizar o impacto da presença de tais fatores. Desta forma uma solução sub-ótima, mas que apresenta pouca variação ante a presença de ruídos e incertezas, pode ser melhor do que uma solução ótima que apresente uma alta variabilidade em sua vizinhança.

Um algoritmo de otimização multiobjetivo deve necessariamente apresentar soluções factíveis e que apresentem uma boa convergência e dispersão das soluções encontradas, utilizando recursos computacionais e de tempo adequados. Um algoritmo de otimização robusta multiobjetivo busca, além da factibilidade, convergência e dispersão, soluções que apresentem a menor sensibilidade a fatores de ruídos e incertezas.

Uma forma de medir a performance de algoritmos é por meio das funções de teste ou funções de *benchmark*, que são problemas artificiais com uma formulação matemática bem definida e que procuram apresentar ao otimizador diversos desafios. A ideia principal que um algoritmo de otimização que apresente uma boa performance em problemas de *benchmark* também apresentará uma boa performance em problemas reais. A literatura especializada apresenta uma grande variedade de problemas [80, 64, 43, 17]. Este trabalho irá focar sua atenção na análise em três conjuntos de problemas teste: ZDT [171], DTLZ [35] e WFG [60]. A escolha das funções testes se justifica por sua ampla utilização na

literatura especializada. Uma característica comum a estes problemas é que um vetor no espaço dos objetivos sempre pode ser escrito como  $\mathbf{x} = (\mathbf{x}_p, \mathbf{x}_d)$ , onde  $\mathbf{x}_p$  é um vetor com  $M - 1$  coordenadas, responsável pelo posicionamento <sup>1</sup> de  $F(\mathbf{x})$  no espaço dos objetivos enquanto que  $\mathbf{x}_d$  é um vetor com  $N - M + 1$  coordenadas, responsável pela convergência de  $F(\mathbf{x})$ .

## 4.1 ZDT

Em Zitzler et al. [171], os autores destacam dois fatores que caracterizam os desafios a serem apresentados a um algoritmo de otimização multiobjetivo: A cobertura da Fronteira Pareto e a manutenção da diversidade da população. Neste artigo é apresentada uma família de funções de teste, chamada de ZDT, formada por seis funções de teste para problemas com apenas dois objetivos, sendo que na função ZDT5 o espaço de decisão apresenta a codificação binária, enquanto que os outros problemas teste as variáveis de decisão pertencem ao intervalo  $[0, 1]$ . Desta forma, em todas as funções,  $\mathbf{x}_p$  é composto por apenas uma coordenada, que corresponde ao valor do objetivo  $f_1(\mathbf{x}) = x_1$ . O segundo objetivo  $f_2(\mathbf{x})$  é definido por meio de diferentes expressões envolvendo  $\mathbf{x}_p$  e  $\mathbf{x}_d$ , sendo responsável pela convergência e distribuição da população na Fronteira Pareto dos problemas. O vetor  $\mathbf{x}_d$  tem um tamanho variável (29 posições para ZDT1 a ZDT3, 9 posições para ZDT4 e ZDT6 e 10 para ZDT5). Os problemas apresentam Fronteira Pareto convexa (ZDT1 e ZDT4), concava (ZDT2 e ZDT6) e desconexa (ZDT3). Por se tratar de problemas com apenas dois objetivos, o foco principal deste conjunto de problemas está concentrado na convergência das soluções obtidas na Fronteira Pareto.

## 4.2 DTLZ

A família de funções de teste DTLZ aparece em dois relatórios técnicos idênticos em 2001 (*Scalable Test Problems for Evolutionary Multi-Objective Optimization*) publicados no *Technical Report No. 112* da *Swiss Federal Institute of Technology (ETH)* e no *KanGAL Report Number 2001001* do *Indian Institute of Technology Kanpur*, em um artigo do congresso *IEEE Congress on Evolutionary Computation - CEC* de 2002 (*Scalable multi-objective optimization test problems*) e como um artigo da revista *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications* no ano de 2005 sob o título *Scalable Test Problems for Evolutionary Multiobjective Optimization*. O artigo apresentado no CEC é uma versão resumida dos relatórios técnicos, com a retirada do problema DTLZ5, DTLZ8 e DTLZ9 e a reordenação dos problemas restantes <sup>2</sup> A diferença

<sup>1</sup> Esta parte do vetor é responsável principalmente pelo posicionamento de  $F(\mathbf{x})$ , mas também influencia na norma de  $F(\mathbf{x})$ . Além disto, esta parte é a única responsável pelo conflito entre os objetivos.

<sup>2</sup> Os problemas DTLZ5 a DTLZ7 apresentados no congresso correspondem aos problemas DTLZ6 a DTLZ8 dos relatórios técnicos. Não houve alteração nos problemas DTLZ1 a DTLZ4 em todos os

entre os relatórios técnicos e o artigo de 2005 é a apresentação, no artigo de 2005, do problema cometa (*Comet Problem*) no apêndice. Este problema estava presente como uma seção no desenvolvimento das ideias apresentadas nos relatórios técnicos (seção 4.6), que foi inteiramente transportada para o apêndice. Este trabalho irá considerar apenas a versão apresentada na revista *Evolutionary Multiobjective Optimization* em 2005, por se tratar da versão mais recente e por apresentar todos os problemas discutidos nos outros trabalhos. Neste trabalho, os autores destacam as seguintes características desejáveis para um problema de teste:

- Os problemas de teste devem apresentar obstáculos (controláveis) que dificultem a convergência e a distribuição das soluções na Fronteira Pareto do problema, e que apresentem dificuldades similares a problemas reais;
- Os problemas devem ser escaláveis para qualquer número de variáveis de decisão e de objetivos;
- Os problemas devem ser simples de construir, sua solução deve ser conhecida, bem como a localização espacial das soluções no espaço dos objetivos.

Os problemas DTLZ1 a DTLZ6 são problemas de otimização multiobjetivo escaláveis para qualquer número de objetivos, sendo desta maneira adequados também para teste em algoritmos de otimização com muitos objetivos. Estes problemas têm uma solução bem definida, a saber,  $x_i \in [0, 1]$  para  $x_i \in \mathbf{x}_p$  e  $x_j = 0.5$  para  $x_j \in \mathbf{x}_d$ . Em todos os problemas, a Fronteira Pareto está localizada no primeiro ortante<sup>3</sup> do espaço dos objetivos e é bem simples: um plano, uma esfera, uma curva ou um simplex. A formulação geral destes problemas é uma função  $F : [0, 1]^N \subset \mathbb{R}^N \rightarrow \mathbb{R}^M$ ,  $\arg \min F(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_M(\mathbf{x}))$ , que pode ser escrita como

$$F(\mathbf{x}) : \mathbb{R}^N \rightarrow \mathbb{R}^M, F(\mathbf{x}) = F_p(\mathbf{x})F_d(\mathbf{x}) \quad (4.1)$$

No problema DTLZ1, a função  $F_p(\mathbf{x})$  é definida como

$$F_p(\mathbf{x}) = \begin{cases} f_{p,1}(\mathbf{x}) = x_1 x_2 \cdots x_{M-1} \\ f_{p,2}(\mathbf{x}) = x_1 x_2 \cdots (1 - x_{M-1}) \\ \vdots \\ f_{p,M-1}(\mathbf{x}) = x_1 (1 - x_2) \\ f_{p,M}(\mathbf{x}) = 1 - x_1 \end{cases} \quad (4.2)$$

---

trabalhos apresentados.

<sup>3</sup> O primeiro ortante é o conjunto de pontos  $\mathbf{x} = (x_1, \dots, x_M)$  do espaço  $M$ -dimensional com  $x_j \geq 0$ ,  $j = 1, \dots, M$ . Em  $\mathbb{R}^2$  o primeiro ortante é o primeiro quadrante, no espaço  $\mathbb{R}^3$  o primeiro ortante é o primeiro octante e assim por diante.

que associa cada ponto do hiperplano  $[0, 1]^N$  do espaço de decisão  $\mathbb{R}^N$  à porção do hiperplano  $f_{p,1} + f_{p,2} + \dots + f_{p,M} = 1$  no primeiro ortante do espaço de decisão  $\mathbb{R}^M$ . Esta função determina a localização espacial dos pontos no espaço dos objetivos e devido às características geométricas deste hiperplano os objetivos estão em conflito, o que é a principal característica de um problema de otimização multiobjetivo. Uma distribuição adequada dos componentes  $x_i$  de  $\mathbf{x}_p$  faz com que estes pontos se distribuam neste hiperplano, garantindo assim uma boa dispersão das soluções na Fronteira Pareto. Observe que um problema com  $M$  objetivos faz uso de um vetor  $M - 1$  componentes.

Para este problema, a convergência dos pontos para a Fronteira Pareto é definida pela função  $F_d(\mathbf{x}) = \frac{1}{2}(1 + g(\mathbf{x}))$  sendo

$$g(\mathbf{x}) = 100 \left( |\mathbf{x}_d| + \sum_{x_i \in \mathbf{x}_d} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)) \right) \quad (4.3)$$

Para  $x_i = 0.5$  com  $x_i \in \mathbf{x}_d$  temos que  $g(\mathbf{x}) = 0$  e portanto a Fronteira Pareto deste problema é a porção do hiperplano  $f_1 + f_2 + \dots + f_M = \frac{1}{2}$  no primeiro ortante de  $\mathbb{R}^M$ . Exclusivamente na equação (4.3)  $|\mathbf{x}_d|$  representa o número de coordenadas do vetor  $\mathbf{x}_d$ , e não a sua norma.

Nos problemas DTLZ2 a DTLZ3 a função  $F_p(\mathbf{x})$  utiliza o sistema de coordenadas esféricas no espaço  $M$ -dimensional para determinar a posição dos pontos no espaço dos objetivos por meio da equação

$$F_p(\mathbf{x}) = \begin{cases} f_{p,1}(\mathbf{x}) = \cos(x_1\pi/2) \cos(x_2\pi/2) \dots \cos(x_{M-2}\pi/2) \cos(x_{M-1}\pi/2) \\ f_{p,2}(\mathbf{x}) = \cos(x_1\pi/2) \cos(x_2\pi/2) \dots \cos(x_{M-2}\pi/2) \sin(x_{M-1}\pi/2) \\ f_{p,3}(\mathbf{x}) = \cos(x_1\pi/2) \cos(x_2\pi/2) \dots \sin(x_{M-2}\pi/2) \\ \vdots \\ f_{p,M-1}(\mathbf{x}) = \cos(x_1\pi/2) \sin(x_2\pi/2) \\ f_{p,M}(\mathbf{x}) = \sin(x_1\pi/2) \end{cases} \quad (4.4)$$

Para  $x_i \in [0, 1]$ ,  $i = 1, \dots, M - 1$  a função corresponde à superfície de uma esfera no primeiro ortante do espaço  $\mathbb{R}^M$ . Mais uma vez as características geométricas desta superfície fazem com que os objetivos sejam conflitantes e uma distribuição adequada dos vetores  $\mathbf{x}_p$  garante uma boa dispersão destes pontos na Fronteira Pareto. No problema DTLZ4, as variáveis  $x_i$  presente nas equações (4.4) são substituídas por  $x_i^\alpha$  de modo a introduzir um viés e dificultar a distribuição espacial dos pontos no espaço dos objetivos. Os autores sugerem  $\alpha = 100$ . Nos problemas DTLZ5 e DTLZ6 as variáveis  $x_i$  presente nas

equações (4.4) são substituídas por  $\frac{\pi}{4(1+g(\mathbf{x}))}(1 + 2g(\mathbf{x})x_i)$  sendo

$$g(\mathbf{x}) = \sum_{x_i \in \mathbf{x}_p} (x_i - 0.5)^2 \quad (4.5)$$

no problema DTLZ5 e

$$g(\mathbf{x}) = \sum_{x_i \in \mathbf{x}_p} x_i^{0.1} \quad (4.6)$$

no problema DTLZ6. Esta mudança faz com que a Fronteira Pareto destes problemas seja uma curva contida em uma esfera no primeiro ortante do espaço dos objetivos em problemas com três objetivos.

A convergência dos pontos para a Fronteira Pareto é definida pela função  $F_d(\mathbf{x}) = 1 + g(\mathbf{x})$  sendo  $g(\mathbf{x})$  definida pela equação (4.3) no problemas DTLZ3, pela equação (4.5) no problemas DTLZ2, DTLZ4 e DTLZ5 e pela equação (4.6) nos problemas DTLZ6.

Os problemas DTLZ7 a DTLZ9 não utilizam o sistema de coordenadas esféricas no espaço  $M$ -dimensional. O problema DTLZ7 apresenta uma formulação simples para os objetivos 1 a  $M - 1$ :  $f_i(\mathbf{x}) = x_i$  para  $1 \leq i \leq M - 1$ . O último objetivo  $f_M(\mathbf{x})$  é o único que depende das demais variáveis do problema. Este problema apresenta  $2^{M-1}$  regiões desconexas na Fronteira Pareto.

Os problemas DTLZ8 e DTLZ9 são os únicos que apresentam restrições de desigualdade desta família, sendo que o problema DTLZ8 apresenta  $M$  restrições e o DTLZ9 apresenta  $M - 1$  restrições. A Fronteira Pareto no espaço com três objetivos do problema DTLZ8 é composta por um segmento de reta e uma superfície plana de formato triangular e a Fronteira Pareto do problema DTLZ9 é similar à Fronteira Pareto do problema DTLZ5.

Nos problemas DTLZ1 a DTLZ7 o espaço de decisão tem  $N = M - 1 + k$  variáveis, sendo as primeiras  $M - 1$  variáveis responsáveis pela localização espacial dos pontos no espaço dos objetivos e as  $k = N - M + 1$  últimas variáveis responsáveis pela convergência dos pontos na Fronteira Pareto. Desta forma, nestes problemas um vetor no espaço de decisão pode ser escrito como  $\mathbf{x} = (\mathbf{x}_p, \mathbf{x}_d)$ , sendo  $\mathbf{x}_p$  a porção do vetor responsável pela localização espacial dos pontos no espaço dos objetivos e  $\mathbf{x}_d$  a porção do vetor responsável pela convergência. Nos problemas DTLZ8 e DTLZ9 os autores sugerem  $N = 10M$  variáveis. Os valores de  $k$  sugeridos são  $k = 5$  para o problema DTLZ1,  $k = 10$  para os problemas DTLZ2 a DTLZ6 e  $k = 20$  para o problema DTLZ7. O conjunto Pareto Ótimo para os problemas DTLZ1 a DTLZ7 é  $x_i \in [0, 1]$  para  $x_i \in \mathbf{x}_p$  e  $x_j = 0.5$  para  $x_j \in \mathbf{x}_d$  nos problemas DTLZ1 a DTLZ5 e  $x_j = 0$  para  $x_j \in \mathbf{x}_d$  no problema DTLZ7. Nos problemas DTLZ8 e DTLZ9 a solução no espaço de decisão não é apresentada.

### 4.3 WFG

Huband et al. [60] dividem as características desejáveis aos problemas de *benchmark* para otimização multiobjetivo em duas categorias: as características gerais da aplicação que define o problema de otimização multiobjetivo, chamadas de *Fitness landscape* e no formato da Fronteira Pareto, chamadas de *pareto optimal front geometry*.

Neste artigo os autores destacam diversas características presentes em problemas de otimização multi-objetivo. Um dos itens analisados é a importância da dimensão do espaço de decisão e dos objetivos. Além de ser bem definido para qualquer número de objetivos, o problema de otimização também deve apresentar a possibilidade de escalabilidade do número de variáveis do espaço de decisão. Um problema com mais variáveis de decisão do que objetivos de maneira geral apresenta mais dificuldades ao otimizador.

Para um vetor  $\mathbf{x} = (x_1, \dots, x_N)$  no espaço de decisão, uma variável qualquer  $x_i$  é classificada de duas formas:  $x_i$  é um parâmetro de distância se sua variação produzir uma nova solução  $\mathbf{y}$  que altera a relação de dominância entre  $F(\mathbf{y})$  e  $F(\mathbf{x})$ . Caso contrário,  $x_i$  é um parâmetro de posição. Se uma variável  $x_i$  apresenta um mesmo valor ótimo  $x_i^*$  para todos os objetivos, então esta variável é separável. Caso contrário,  $x_i$  é não-separável. Se toda variável de um objetivo  $f_i(\mathbf{x})$  for separável, então este objetivo é separável. Se todos os objetivos forem separáveis, então o problema  $F(\mathbf{x})$  é separável.

As soluções  $\mathbf{x}^* = (x_1^*, \dots, x_N^*)$  do problema no espaço de decisão são classificadas de acordo com a localização de  $x_i^*$  no intervalo  $[L_i, U_i]$  onde esta variável está definida: Se  $x_i^*$  está próxima dos extremos  $L_i$  ou  $U_i$ , então  $x_i^*$  é um parâmetro externo (*external parameter*). Caso contrário, se  $x_i^*$  se localiza próximo ao centro do intervalo  $[L_i, U_i]$ , então  $x_i^*$  é um parâmetro central (*medial parameter*).

Em relação à convergência, um problema pode ser classificado pela presença de vários ótimos locais, o que caracteriza um problema multimodal, ou um único ótimo global no espaço de busca, o que caracteriza problemas mono-modais. Um tipo específico de problema multimodal é o problema deceptivo. Em um problema deceptivo as soluções sub-ótimas conduzem a população para uma região distante daquela onde se localiza o ótimo global.

Um último aspecto a ser analisado é o formato da Fronteira Pareto do problema. Em um problema com  $M$  objetivos a Fronteira Pareto é, de maneira geral, uma superfície de dimensão  $M - 1$ . Se a dimensão da Fronteira Pareto for menor do que  $M - 1$  então o problema é degenerado. Esta superfície pode ser côncava, convexa, plana ou uma mistura destes formatos. Esta superfície pode também ser conexa<sup>4</sup> ou desconexa.

Neste artigo, os autores apresentam diversas recomendações para a construção de

<sup>4</sup> Uma superfície  $S$  é conexa se, dados dois pontos quaisquer  $A$  e  $B$  em  $S$ , sempre existe um caminho  $c$  contido em  $S$  conectando estes pontos.

problemas teste, das quais destacamos:

- não se deve utilizar parâmetros externos ou centrais;
- a dimensão do espaço de decisão e dos objetivos deve ser ajustável;
- o espaço de busca e de objetivos devem ser dissimilares, ou seja, as variáveis no espaço de busca devem estar contidas em intervalos de diferentes tamanhos (*dissimilar parameter domains*), bem como as soluções no espaço dos objetivos (*dissimilar tradeoff ranges*);
- os problemas devem ter a solução ótima conhecida;
- os problemas devem apresentar uma riqueza de formato de Fronteira Pareto (concava, convexa, plana, conexa, desconexa etc)

Balizado por estas recomendações, os autores apresentam um conjunto de nove funções teste (WFG1 a WFG9). Partindo de um vetor de parâmetros  $\mathbf{z}$ , uma sequência de transformações é aplicada de modo a se obter um outro vetor  $\mathbf{x}$  que agrega as características desejadas. O problema é então definido pela minimização dos objetivos  $f_i(\mathbf{x})$ ,  $1 \leq i \leq M$ . O vetor  $\mathbf{z}$  tem  $k + l = N \geq M$  posições, sendo as primeiras  $k$  posições responsáveis pelo posicionamento de  $F(\mathbf{x})$  no espaço dos objetivos e as últimas  $l$  posições responsáveis pela distância de  $F(\mathbf{x})$  até a Fronteira Pareto do problema. O vetor transformado  $\mathbf{x}$  tem  $M$  posições, sendo as primeiras  $M - 1$  coordenadas responsáveis pelo posicionamento de  $F(\mathbf{x})$  no espaço dos objetivos e a última posição responsável pela distância de  $F(\mathbf{x})$  até a Fronteira Pareto do problema. Desta forma, esta classe de problemas pode ser representada pela sequência de aplicações  $\mathbf{Z} \xrightarrow{t(\mathbf{z})} \mathbf{X} \xrightarrow{F(\mathbf{x})} \mathbf{Y}$ , sendo  $\mathbf{Z}$  o espaço de decisão,  $\mathbf{X}$  o espaço dos parâmetros do problema e  $\mathbf{Y}$  o espaço dos objetivos, com  $0 \leq z_i \leq z_{i,\max}$  e  $0 \leq x_i \leq 1$ .

Todos os problemas têm solução (não externa e não central) conhecidas. Os parâmetros de posição não estão restritos ao número de objetivos e apresentam uma rica variedade de desafios para o otimizador.

## 4.4 Limitações das funções

A principal limitação da família ZDT é estar restrita a dois objetivos. Entretanto, por apresentar um grande número de variáveis responsáveis pela convergência da população e por ser de fácil implementação, este conjunto de funções teste ainda é válido para avaliar algoritmos com dois objetivos.

Na família DTLZ os problemas mais utilizados em otimização multiobjetivo são os DTLZ1 a DTLZ4. Estes problemas são escaláveis a qualquer número de objetivos, mas apresentam pouca variedade de formatos de sua Fronteira Pareto, a saber, um plano

para o problema DTLZ1 e uma esfera para os demais. Além disso, o número de variáveis relacionadas ao posicionamento das soluções  $F(\mathbf{x})$  no espaço dos objetivos é restrito a  $M - 1$ . Os autores sugerem uma forma de aumentar a dificuldade deste aspecto dos problemas por meio da substituição do valor nominal da variável  $x_i$  ( $1 \leq i \leq M - 1$ ) pela meta-variável  $y_i$  dada pelo valor médio de  $q$  variáveis utilizando a expressão

$$y_i = \frac{1}{q} \sum_{k=(i-1)q+1}^{iq} x_k \quad (4.7)$$

o que faz com que a localização espacial dos pontos no espaço de decisão dependa de um número maior de variáveis. Entretanto, esta estratégia já não oferece um desafio significativo para os algoritmos de otimização multiobjetivo atuais. A convergência destes problemas é controlada pela função  $g(\mathbf{x})$  definida pelas equações (4.3) (para os problemas DTLZ1 e DTLZ3) e (4.5) (para os problemas DTLZ2, e DTLZ4).

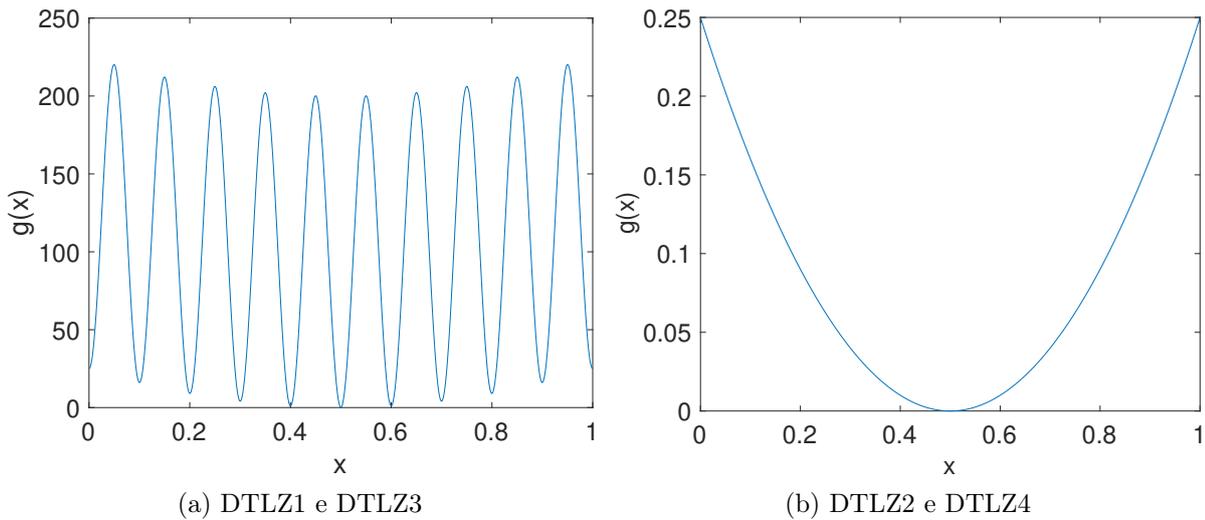


Figura 16 – Função  $g(\mathbf{x})$  utilizada nos problemas DTLZ1 a DTLZ4 restrita a uma variável  $0 \leq x \leq 1$

As figuras 16a e 16b apresentam o gráfico das funções  $g(\mathbf{x})$ , restritas a uma única variável  $\mathbf{x}$ . A equação (4.3) utilizada nos problemas DTLZ1 e DTLZ3 apresenta um problema multimodal, enquanto que a equação (4.5), utilizada nos problemas DTLZ2 e DTLZ4 é monomodal e ambas já não apresentam um desafio significativo para os algoritmos de otimização multiobjetivo mais recentes. Os problemas DTLZ5 e DTLZ6, que têm Fronteira Pareto degenerada apresentam inconsistências em problemas com mais de 3 objetivos (como apontado em Huband et al. [60]) O problema DTLZ7 tem uma formulação muito simples, a saber,  $f_i(\mathbf{x}) = x_i$ ,  $1 \geq i \geq M - 1$ , sendo  $f_M(\mathbf{x})$  o único objetivo a apresentar uma expressão algébrica mais elaborada. Além disso, apresenta como solução ótima um valor externo ( $x_i = 0, M \leq i \leq N$ ). Os problemas DTLZ8 e DTLZ9

são parcialmente degenerados e seus resultados são de difícil interpretação em dimensões elevadas, além de não apresentarem uma solução conhecida.

A família de problemas WFG foi cuidadosamente elaborada, balizada pela lista de virtudes e falhas levantadas após uma extensa e rigorosa análise dos trabalhos apresentados até então. Entretanto, esta classe de problemas também apresenta alguns pontos fracos. Por exemplo, uma característica presente no problema WFG1 é a ideia de regiões planas (*flat regions*) no espaço dos objetivos. Uma *flat region* é uma região onde pequenas perturbações das variáveis não provoca alterações no valor dos objetivos, que corresponde justamente à ideia de otimização robusta. A presença do ótimo global em uma região plana pode fazer que algoritmos com performance distinta apresentem resultados próximos. A falta de significado geométrico das transformações também dificulta a análise dos resultados obtidos, que ficam restritos às métricas de avaliação existentes, que apresentam muitas limitações técnicas. Outra limitação presente nesta classe de problemas, e também presente nas famílias ZDT e DTLZ é a ausência de problemas dedicados para a otimização robusta, bem como o tratamento de restrições de desigualdade e igualdade, exceto os problemas DTLZ8 e DTLZ9 que apresentam restrições de desigualdade, mas sem possibilidade de personalização. Uma outra questão pertinente, que é consequência da complexidade destas funções, é sua implementação e interpretação, que não é tão simples quanto as outras funções analisadas.

## 4.5 Uma nova família de funções de teste

Esta seção apresenta um novo conjunto de funções de teste. Esta proposta procura seguir as seguintes diretrizes:

1. Apresentar liberdade de escolha do número de variáveis no espaço de decisão e de objetivos e que tenham sua solução bem definida tanto no espaço de decisão quanto no espaço dos objetivos;
2. apresentar um conjunto de restrições de desigualdade e de igualdade que sejam fáceis de interpretar e de identificar de sua validade e sua violação;
3. apresentar um conjunto de parâmetros em cada função que ajuste, de forma independente, os desafios apresentados ao otimizador quanto à convergência e a cobertura da Fronteira Pareto;
4. apresentar um conjunto de funções de teste que sejam válidas para o teste de algoritmos de otimização robusta, com solução bem definida e de fácil interpretação;
5. apresentar uma rica variedade de formatos para a Fronteira Pareto.

Juntamente com este novo conjunto de funções de teste, uma nova métrica de convergência é proposta. As métricas de convergência disponíveis são dependentes de uma amostra particular da Fronteira Pareto do problema (tais como o *Generational Distance* (GD) [148] ou a *Inverted Generational Distance* (IGD) [23]) ou têm um custo computacional muito elevado (como o *Hypervolume indicator* (HV)[170]). A nova métrica de convergência proposta tem como objetivo apresentar uma medida de convergência aplicada em problemas teste que não dependa de uma amostra particular da Fronteira Pareto.

Os problemas de teste propostos procuram a minimização da função

$$F : \mathbb{R}^N \rightarrow \mathbb{R}^M$$

$$F(\mathbf{x}) = F_p(\mathbf{x})F_d(\mathbf{x}) \quad (4.8)$$

Nestes problemas, um vetor  $\mathbf{x}$  no espaço de decisão  $\mathbb{R}^N$  pode ser escrito da forma  $\mathbf{x} = (\mathbf{x}_p, \mathbf{x}_d) \in \mathbb{R}^N = \mathbb{R}^R \times \mathbb{R}^S$ , sendo  $\mathbf{x}_p = (x_1, \dots, x_R) \in \mathbb{R}^R$  com  $-1 \leq x_i \leq 1$ ,  $1 \leq i \leq R$  um vetor com  $R$  posições, responsável pelo posicionamento dos pontos  $F(\mathbf{x})$  no espaço dos objetivos e  $\mathbf{x}_d = (x_1, \dots, x_S) \in \mathbb{R}^S$  com  $0 \leq x_j \leq 1$ ,  $1 \leq j \leq S$  um vetor com  $S$  posições, responsável pela convergência de  $F(\mathbf{x})$ . Os detalhes das funções  $F_p(\mathbf{x})$  e  $F_d(\mathbf{x})$  serão discutidos nas seções seguintes.

#### 4.5.1 A função $F_p(\mathbf{x})$

A função  $F_p(\mathbf{x})$  é responsável pela posição relativa dos pontos no espaço dos objetivos e pelo conflito entre os objetivos. A definição básica desta função é dada pela equação (4.4), que descreve em coordenadas esféricas a superfície de uma hipersfera de raio 1 no primeiro ortante do espaço  $\mathbb{R}^M$  e depende de  $M - 1$  variáveis  $x_i$ . A escolha desta equação se justifica pelo fato de que esta equação descreve a localização espacial de pontos no espaço dos objetivos com simplicidade e precisão. Entretanto são apresentadas duas modificações nesta expressão que tem como objetivo aumentar o número de variáveis e a diversidade de formato desta superfície.

Uma forma de se aumentar a dificuldade apresentada para o otimizador em encontrar a melhor distribuição dos pontos no espaço dos objetivos é substituir a variável  $x_i$  da equação  $F_p(\mathbf{x})$  pela meta-variável  $y_i$  definida pela expressão

$$y_i = \frac{1}{q+t} \left| \sum_{j=(i-1)q+1}^{iq+t} x_j \right| \quad (4.9)$$

que corresponde à média de  $q+t$  valores contidos no conjunto de variáveis  $\{x_{(i-1)q+1}, \dots, x_{iq+t}\}$  do espaço de decisão original. Observe que, a diferença entre as meta-variáveis apresentadas

nas equações (4.7) e (4.9) é a soma do parâmetro  $t$  no limite superior do somatório.

$$\begin{aligned}
 \mathbf{x} &= \left( \overbrace{x_1, \dots, x_{q+1}, \dots, x_{q+t}}^{y_1}, \overbrace{x_{q+t+1}, \dots, x_{2q}}^{y_2}, \overbrace{x_{2q+1}, \dots, x_{2q+t}, x_{2q+t+1}, \dots, x_{3q+t}}^{y_3}, \dots \right. \\
 &\quad \left. \dots, \overbrace{x_{(M-2)q+t+1}, \dots, x_{(M-1)q+t}}^{y_{M-1}} \right) \\
 y_1 &= \frac{1}{q+t} \left| \sum_{j=1}^{q+t} x_j \right|, \\
 y_2 &= \frac{1}{q+t} \left| \sum_{j=q+1}^{2q+t} x_j \right|, \\
 &\quad \vdots \\
 y_{M-1} &= \frac{1}{q+t} \left| \sum_{j=(M-2)q+1}^{(M-1)q+t} x_j \right|
 \end{aligned} \tag{4.10}$$

Ao contrário da meta-variável (4.7) proposta por Deb para a família DTLZ, na meta-variável apresentada na equação (4.9) cada sub-intervalo onde é calculada a média compartilha  $t$  elementos com o subintervalo anterior e  $t$  elementos com o subintervalo seguinte, aumentando a dependência entre as variáveis do espaço de decisão. Entretanto é necessário que  $2t + 1 < q$  para que cada sub-intervalo apresente pelo menos uma variável  $x_i$  independente dos intervalos vizinhos de modo que exista pelo menos uma solução que alcance qualquer região da Fronteira Pareto. Assim o subespaço  $\mathbb{R}^R$  responsável pelo posicionamento dos pontos no espaço de decisão terá  $(M - 1)q + t$  variáveis. A distribuição das meta-variáveis  $y_i$  no vetor de parâmetros  $\mathbf{x}$  é apresentada nas equações (4.10).

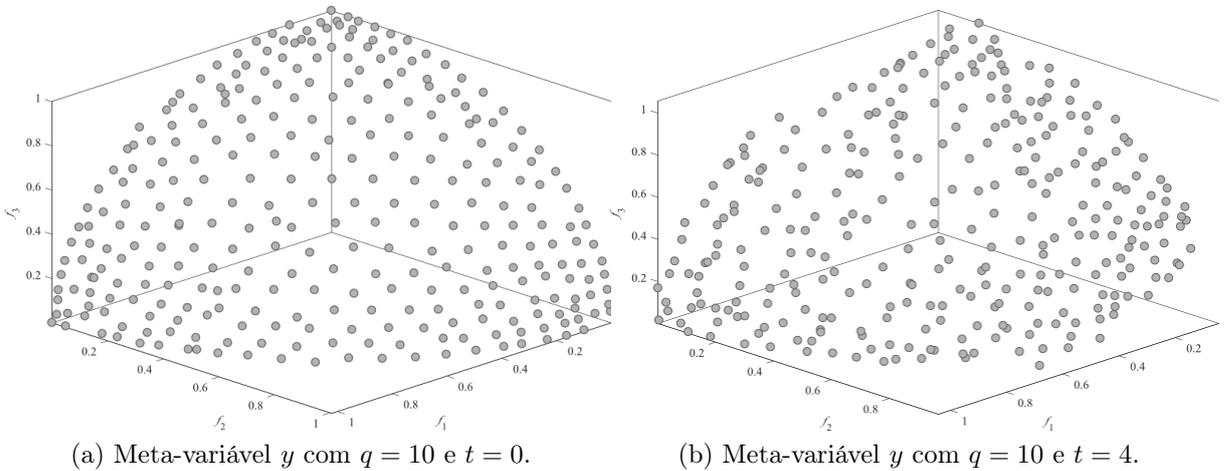


Figura 17 – Diferentes valores do parâmetro  $t$  da meta-variável  $y$  aplicada no problema DTLZ2

A figura 17 ilustra o efeito do uso das meta-variáveis  $\mathbf{y}_i$  em um problema de otimização multiobjetivo. Foi selecionado o problema DTLZ2 com 3 objetivos e utilizada a meta variável  $\mathbf{y}$  apresentada na equação (4.9) utilizando o parâmetro  $q = 10$ . Na figura 17a foi utilizado o parâmetro  $t = 0$ , que corresponde ao uso da meta variável  $y$  definida pela equação (4.7), enquanto que na figura 17b foi utilizado o parâmetro  $t = 4$ . Em ambas situações foi utilizado o algoritmo NSGA-III com parâmetros idênticos. Na comparação entre figuras 17a e 17b é perceptível o efeito da degradação da distribuição espacial das soluções com o uso da meta-variável proposta. O algoritmo utilizado nestes problemas bem como em outros experimentos realizados neste trabalho compõe a plataforma platEMO [144] cuja cópia pode ser obtida no endereço <http://bimk.ahu.edu.cn/index.php?s=/Index/Software/index.html>

O formato desta superfície pode ser controlado pela alteração da norma dos pontos. Se  $\mathbf{x} = (x_1, \dots, x_M)$  é um ponto do espaço  $\mathbb{R}^M$ , a  $p$ -norma  $\|\mathbf{x}\|_p$  (também chamada de norma  $\ell_p$ ) de  $\mathbf{x}$  é dada pela equação

$$\|\mathbf{x}\|_p = \left( \sum_{i=1}^M |x_i|^p \right)^{1/p} \quad (4.11)$$

com  $p \geq 1$  e  $|x_i|$  o valor absoluto de  $x_i$ <sup>5</sup>. Se  $p = 2$  a  $p$ -norma é chamada de norma Euclidiana e se  $p = 1$  ela é chamada de norma Manhattan (ou *taxicab norm*). A norma infinita (ou norma de Tchebycheff) do vetor  $\mathbf{x} \in \mathbb{R}^M$ , denotada por  $\|\mathbf{x}\|_\infty$ , é definida como  $\|\mathbf{x}\|_\infty = \max |x_i|, 1 \leq i \leq M$ . Naturalmente se tem que  $\lim_{p \rightarrow \infty} \|\mathbf{x}\|_p = \|\mathbf{x}\|_\infty$ .

Se  $0 < p < 1$ , a expressão (4.11) não define uma norma, mas uma quasi-norma uma vez que a desigualdade triangular não é satisfeita ([126]). Apesar disso, valores de  $p$  maiores do que zero serão utilizados na equação (4.11) de modo a produzir superfícies de norma (ou quasi-norma) constante. A figura 18 apresenta curvas de norma constante com diferentes valores de  $p$  no espaço  $\mathbb{R}^2$ .

Desta forma a função  $F_p(\mathbf{x})$  é dada pela equação

$$F_p(\mathbf{x}) = \frac{T(\mathbf{x})}{h(\mathbf{x})} \quad (4.12)$$

sendo  $T(\mathbf{x})$  o funcional responsável pela distribuição dos pontos no espaço dos objetivos e definido pela equação (4.13), dependente da meta-variável  $y_i = \frac{1}{q+t} \left| \sum_{j=(i-1)q+1}^{iq+t} x_j \right|$  e

<sup>5</sup> Note que a o valor absoluto de  $x_i$  é a norma  $\ell_2$  de  $x_i$ , i.e.,  $|x_i| = \|x_i\|_2$ .

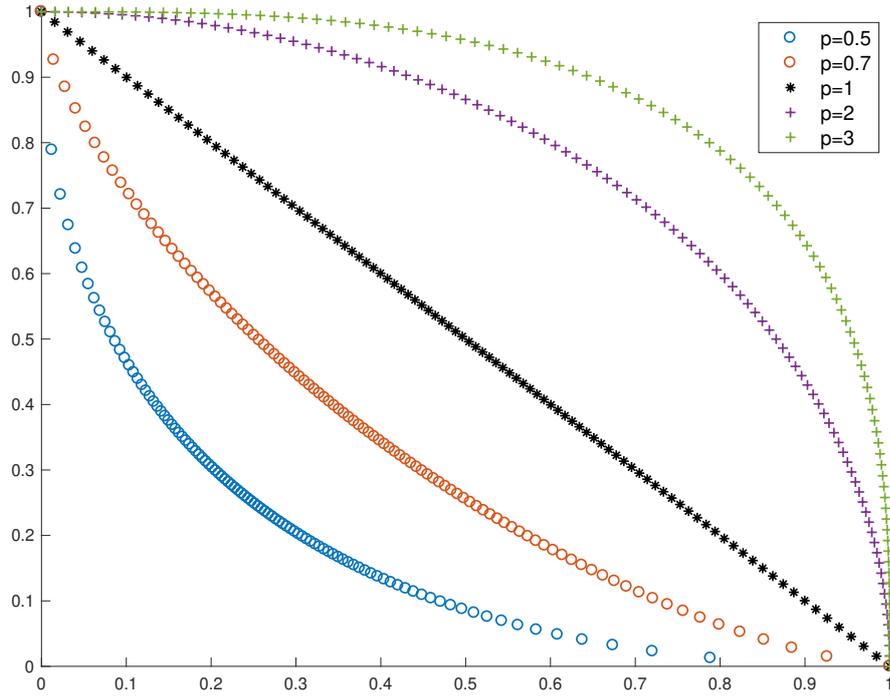


Figura 18 – Pontos com diferentes valores de  $p$ -norma no espaço  $\mathbb{R}^2$ .

$$h(\mathbf{x}) = \|T(\mathbf{x})\|_p.$$

$$T(\mathbf{x}) = \begin{cases} t_1(\mathbf{x}) = \cos(y_1\pi/2) \cos(y_2\pi/2) \dots \cos(y_{M-2}\pi/2) \cos(y_{M-1}\pi/2) \\ t_2(\mathbf{x}) = \cos(y_1\pi/2) \cos(y_2\pi/2) \dots \cos(y_{M-2}\pi/2) \sin(y_{M-1}\pi/2) \\ t_3(\mathbf{x}) = \cos(y_1\pi/2) \cos(y_2\pi/2) \dots \sin(y_{M-2}\pi/2) \\ \vdots \\ t_{M-1}(\mathbf{x}) = \cos(y_1\pi/2) \sin(y_2\pi/2) \\ t_M(\mathbf{x}) = \sin(y_1\pi/2) \end{cases} \quad (4.13)$$

Uma outra característica desejável para problemas de otimização multiobjetivo é a dissimilaridade dos objetivos. De maneira geral, cada objetivo dos problemas de *benchmark* para otimização multiobjetivo tem seus valores ótimos limitados no intervalo  $[0, 1]$ , ou seja  $0 \leq f_i(\mathbf{x}) \leq 1$ , o que dificilmente reflete problemas do mundo real. Alguns autores, como Huband et al. [60] e Cheng et al. [17], apresentam funções com objetivos dissimilares sendo  $0 \leq f_i(\mathbf{x}) \leq \xi$ , com  $\xi$  uma potencia ou um múltiplo de 2. Entretanto estes problemas apresentam apenas objetivos com valores não negativos. Um problema de otimização com

objetivos dissimilares com valores positivos e negativos é apresentado na equação (4.14)

$$D(\mathbf{x}) = \begin{cases} d_1(\mathbf{x}) = 2(2f_1(\mathbf{x}) - 1) \\ d_2(\mathbf{x}) = 4(2f_2(\mathbf{x}) - 1) \\ d_3(\mathbf{x}) = 6(2f_3(\mathbf{x}) - 1) \\ \vdots \\ d_M(\mathbf{x}) = (2M)(2f_M(\mathbf{x}) - 1) \end{cases} \quad (4.14)$$

sendo  $f_i(\mathbf{x})$  o  $i$ -ésimo objetivo obtido após a avaliação de  $F(\mathbf{x}) = F_p(\mathbf{x})F_d(\mathbf{x})$

Note que como  $0 \leq f_i(\mathbf{x}) \leq 1$  então  $-2i \leq d_i(\mathbf{x}) \leq 2i$ . De fato, se  $f_i(\mathbf{x}) = 0$  então  $d_i(\mathbf{x}) = 2i(0 - 1) = -2i$  e se  $f_i(\mathbf{x}) = 1$  então  $d_i(\mathbf{x}) = 2i((2 - 1) - 1) = 2i$ . Observe que a função  $D(\mathbf{x})$  translada a origem do espaço dos objetivos do ponto  $A = (0, \dots, 0)$  para o ponto  $B = (-2, -4, \dots, -2M)$ , mas não altera a concavidade das superfícies.

## 4.5.2 A função $F_d(\mathbf{x})$

Esta função é responsável pela convergência dos pontos na Fronteira Pareto e, em alguns casos, pelo formato da mesma. Esta função faz uso das funções auxiliares  $g(\mathbf{x})$  e  $\phi(\mathbf{x})$ .

### 4.5.2.1 A função $\phi(\mathbf{x})$

A função  $\phi(\mathbf{x})$  tem como objetivo incorporar alguma informação sobre a posição de  $F(\mathbf{x})$  em relação à hiper-diagonal  $\mathbf{d} = (1, 1, \dots, 1)$  no espaço dos objetivos. Para o ponto  $\mathbf{x} \in \mathbb{R}^N$  no espaço de decisão, o ponto  $F_p(\mathbf{x})$  se encontra em uma superfície de  $p$  norma (quasi-norma) constante no espaço dos objetivos, com  $\|F_p(\mathbf{x})\|_p = 1$ . O ângulo  $\varphi$  entre a hiper-diagonal  $\mathbf{d}$  e  $F_p(\mathbf{x})$  é calculado por meio da equação

$$\varphi(\mathbf{x}) = \arccos \left( \frac{\mathbf{d} \cdot F_p(\mathbf{x})^T}{|\mathbf{d}| \|F_p(\mathbf{x})\|} \right) \quad (4.15)$$

Este valor deve ser normalizado no intervalo  $[0, 1]$ . Como o ângulo  $\varphi$  é máximo se o vetor definido por  $F_p(\mathbf{x})$  estiver alinhado com algum vetor da base canônica de  $\mathbf{e}_i = (0, \dots, 1, \dots, 0)$ , ou seja, se  $F_p(\mathbf{x}) = \lambda \mathbf{e}_i$  para algum  $\lambda > 0$ . Neste caso, o ângulo máximo é  $\varphi_{\max} = \arccos \left( \frac{1}{\sqrt{M}} \right)$  e o valor da função  $\phi(\mathbf{x}_d)$  é dado por

$$\phi(\mathbf{x}) = \frac{\varphi(\mathbf{x})}{\varphi_{\max}} \quad (4.16)$$

4.5.2.2 A função  $g(\mathbf{x})$ 

A função  $g(\mathbf{x})$  é responsável pela convergência dos pontos no espaço dos objetivos e apresenta duas versões: uma função deceptiva<sup>6</sup> e uma função robusta.

Uma função deceptiva se caracteriza pela presença de um mínimo global e pelo menos um mínimo local. Neste tipo de problema, a topologia do problema favorece as soluções sub-ótimas, fazendo que o ótimo global seja difícil de ser alcançado. A função  $g(\mathbf{x})$  deceptiva proposta é definida pela equação (4.17) e apresenta, para cada uma das variáveis  $x_i \in \mathbf{x}_d$ , dois mínimos locais (em  $x_i = 0$  e  $x_i = 1$ ) e um mínimo global em  $x_i = 0.5$ , localizado em um vale profundo de amplitude  $2r$ . Para  $0 \leq x_i \leq 0.5 - r$  (e  $0.5 + r \leq x_i \leq 1$ ), a função  $g(\mathbf{x})$  é uma linha conectando os pontos  $(0, 0.5)$  e  $(0.5 - r, 1)$  ( $(0.5 + r, 1)$  e  $(1, 0.5)$  respectivamente). Para  $0.5 - r \leq x_i \leq 0.5 + r$ ,  $g(\mathbf{x})$  é um ciclo completo da função trigonométrica cosseno. Este formato especial tem como motivação esconder a localização do valor ótimo  $x_i$  pela pequena abertura do vale determinado pelo parâmetro  $r$ , conduzindo a população para os ótimos locais localizados no extremo dos intervalos. A função  $g(\mathbf{x})$  deceptiva proposta é definida pela equação (4.17)

$$g(\mathbf{x}) = \sum_{x_i \in \mathbf{x}_d} z(x_i) \quad (4.17)$$

sendo

$$z(x_i) = \begin{cases} 0.5 + \frac{x_i}{1 - 2r} & \text{se } 0 \leq x_i < 0.5 - r \\ \frac{1}{2} \left( \cos \left( \frac{\pi(x_i - 0.5 + r)}{r} \right) + 1 \right) & \text{se } 0.5 - r \leq x_i \leq 0.5 + r \\ 1 - \frac{x_i - 0.5 - r}{1 - 2r} & \text{se } 0.5 + r < x_i \leq 1 \end{cases} \quad (4.18)$$

A figura 19 apresenta o gráfico de  $g(\mathbf{x})$  para uma única variável  $x_i \in [0, 1]$  com o parâmetro  $r$  igual a 0.1, 0.05 e 0.01.

A figura 20 apresenta as soluções obtidas pelo algoritmo MOEA/D utilizando 600 indivíduos em 300 iterações para o problema DTLZ2 utilizando a função  $g(\mathbf{x})$  deceptiva proposta, com parâmetro  $r$  igual a 0.1, 0.05 e 0.01.

A outra versão para a função  $g(\mathbf{x})$  se aplica à otimização robusta. A função  $g(\mathbf{x})$  robusta é a combinação de funções logísticas e trigonométricas. Esta função não apresenta variáveis externas e as soluções robustas estão localizadas no intervalo  $(0.1, 0.3)$ . A solução ótima é  $x_i = 0.6$ <sup>7</sup> mas sua vizinhança é sensível ao ruído apresentado por  $g(\mathbf{x})$ . A função  $g(\mathbf{x})$  robusta é definida pela equação (4.19) e a figura 21 apresenta o gráfico desta função para uma única variável  $x_1 \in [0, 1]$ .

<sup>6</sup> Do latim *deceptus*, participio de *decipio*: apanhar, surpreender, enganar, iludir.

<sup>7</sup> Mais precisamente  $x_i = 0.600066066066066\dots$

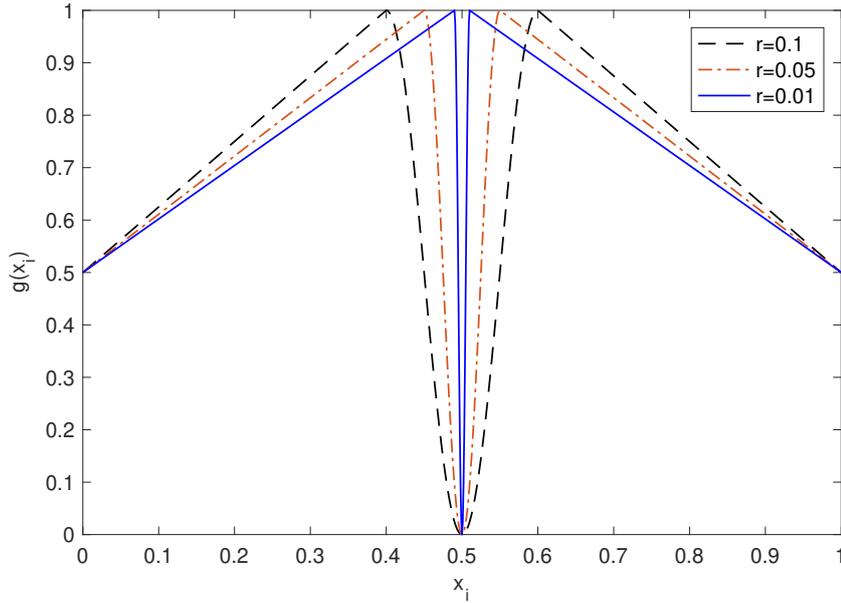


Figura 19 – Função de teste  $g(\mathbf{x})$  em uma variável com  $r = 0.1, 0.05$  e  $0.01$

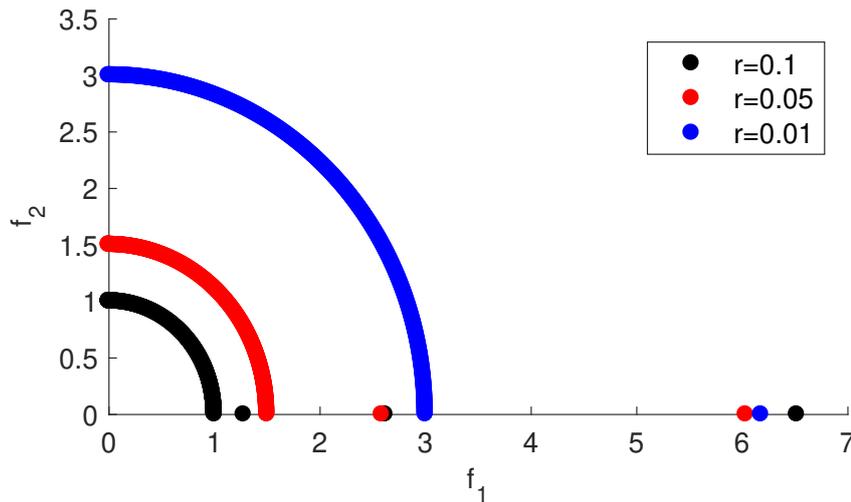


Figura 20 – DTLZ2 com 2 objetivos utilizando a função de teste  $g(\mathbf{x})$  e  $r = 0.1, 0.05$  e  $0.01$

$$g(\mathbf{x}) = \sum_{x_i \in \mathbf{x}_d} \left[ -w(x_i) (y(x_i) - z(x_i)) + \frac{y(x_i) - 1}{2} + e^{-60x_i} + 0.631 \right] \quad (4.19)$$

onde

$$y(x_i) = \frac{1}{1 + e^{-20(x_i - 0.6)}} \quad (4.20)$$

$$z(x_i) = \frac{1}{1 + e^{-20(x_i - 0.7)}} \quad (4.21)$$

$$w(x_i) = \cos(40\pi x_i) \quad (4.22)$$

A figura 22 ilustra a resposta provocada pela presença de incertezas nas variáveis  $\mathbf{x}_d$  desta função. Esta figura apresenta as soluções obtidas com o valor ótimo ( $x_i = 0.6, \in \mathbf{x}_d$

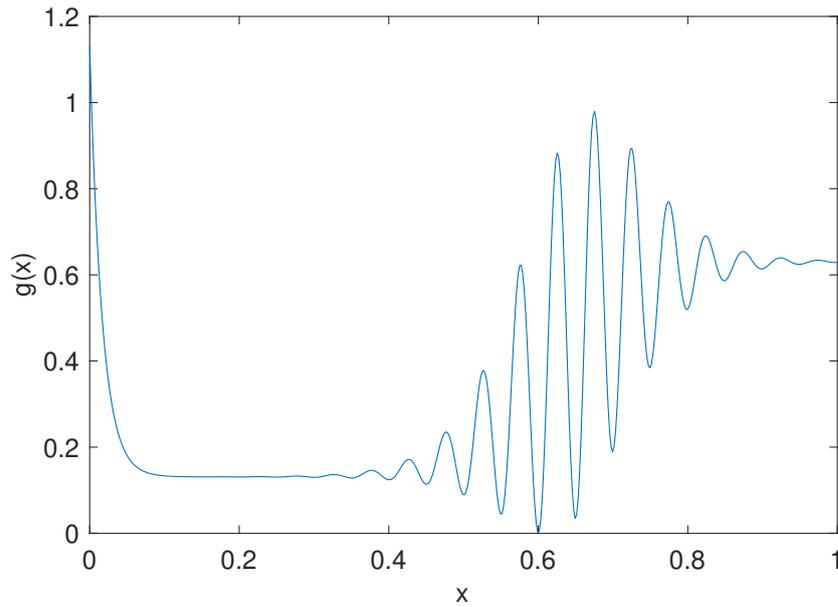


Figura 21 – A função  $g(\mathbf{x})$  robusta restrita a uma variável  $0 \leq x \leq 1$

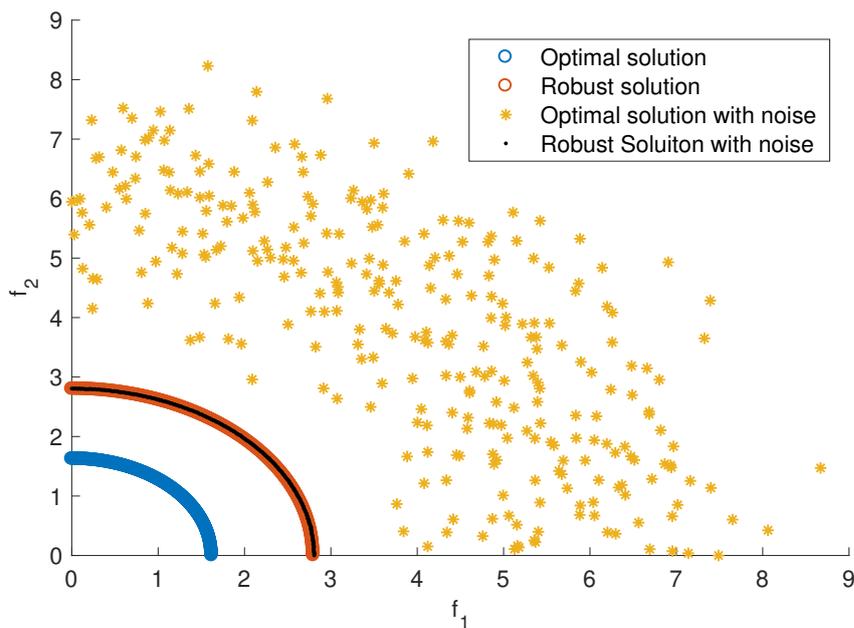


Figura 22 – Efeito da presença de ruído em uma vizinhança dos valores ótimos globais e robustos para a função  $g(\mathbf{x})$  robusta.

- pontos azuis) e robusto ( $x_i = 0.2 \in (0.1, 0.3)$  - pontos vermelhos) no problema DTLZ2 com dois objetivos, utilizando esta função. Para verificar a robustez destas soluções, foi acrescentado um ruído  $\epsilon$  com distribuição uniforme ( $\epsilon \sim U(-0.1, 0.1)$ ) nas soluções ótima e robusta. Em seguida, estas soluções foram avaliadas novamente. A figura mostra que não houve alteração na distribuição e na convergência das soluções robustas (pontos pretos), enquanto que as soluções ótimas com ruído apresentam soluções insatisfatórias (pontos amarelos).

### 4.5.3 Restrições de igualdade e desigualdade

As características do problema de otimização  $F(\mathbf{x})$  proposto permite facilmente a criação de restrições de igualdade e desigualdade a partir da manipulação da função  $\varphi(\mathbf{x})$  definida na equação (4.15), uma vez que esta função permite a localização espacial de pontos no espaço dos objetivos. Por exemplo, pode-se restringir as soluções no espaço dos objetivos que estejam limitados a uma distancia angular  $\theta_i$ , com  $0 \leq \theta_i \leq \varphi_{\max}$  como apresentado nas equações (4.23) a (4.26)

$$\varphi(\mathbf{x}) \geq \theta \quad (4.23)$$

$$\varphi(\mathbf{x}) \leq \theta \quad (4.24)$$

$$\theta_1 \leq \varphi(\mathbf{x}) \leq \theta_2 \quad (4.25)$$

$$\varphi(\mathbf{x}) = \theta \quad (4.26)$$

$$(4.27)$$

Uma possível variação para estas restrições é a alteração do vetor de referência  $\mathbf{d}$  associado à função  $\varphi(\mathbf{x})$ . Por exemplo considere um problema onde o espaço dos objetivos está contido no primeiro ortante do espaço  $\mathbb{R}^M$ . Neste caso, considere o ângulo  $\theta_i$  entre o ponto  $y = F(\mathbf{x})$  e os vetores da base canônica  $e_i = (0, \dots, 1, \dots, 0)$ , com  $1 \leq i \leq M$ . Seja  $\theta_j = \min\{\theta_1, \dots, \theta_M\}$  e  $j$  o objetivo associado a este eixo. A função  $\xi(\mathbf{x}) = j$  associa o vetor  $\mathbf{x}$  ao objetivo  $j$  que está a uma menor distância angular de  $F(\mathbf{x})$ . É fácil ver que  $\xi(\mathbf{x}) = j$  não é uma função injetora, pois para  $F(\mathbf{x}) = (1, \dots, 1)$  se tem que  $\theta_1 = \theta_2 = \dots = \theta_M$ , por exemplo. Em situações como esta, onde  $\min\{\theta_1, \dots, \theta_M\} = \{\theta_{j1}, \dots, \theta_{jk}\}$ , faça  $\xi(\mathbf{x}) = \min\{j1, \dots, jk\}$  e defina a restrição

$$\xi(\mathbf{x}) = j \quad (4.28)$$

### 4.5.4 Funções propostas

As funções auxiliares definidas na seção anterior permitem a criação de diversos problemas de otimização, identificados por **pd-MOP-XX**, sendo **XX** um número inteiro positivo. O que difere cada um deles é essencialmente o formato da Fronteira Pareto. Os problemas são escaláveis a qualquer número de objetivos e utilizam por padrão a função  $g(\mathbf{x})$  deceptiva definida pela equação (4.17). Cada um deles terá uma versão robusta, utilizando a função  $g(\mathbf{x})$  robusta definida pela equação (4.19) e uma versão com objetivos dissimilares. Cada um dos problemas também terá a opção de apresentar qualquer uma das restrições descritas na seção 4.5.3. Todos os problemas propostos são da forma

$$F : \mathbb{R}^N \rightarrow \mathbb{R}^M$$

$$F(\mathbf{x}) = F_p(\mathbf{x})F_d(\mathbf{x}) \quad (4.29)$$

O espaço de decisão  $\mathbb{R}^N = \mathbb{R}^R \times \mathbb{R}^S$  é separável, sendo os vetores do espaço  $\mathbb{R}^R$  responsáveis pela posição relativa de  $F(\mathbf{x})$  no espaço dos objetivos e os vetores de  $\mathbb{R}^S$  responsáveis pela convergência dos pontos na Fronteira Pareto. O espaço  $\mathbb{R}^R$  tem  $r = (M - 1)q + t$  coordenadas, sendo  $M$  o número de objetivos e  $q, t$  os parâmetros da meta variável  $y$  definida pela equação (4.9). Sugerimos  $q = 10$  e  $t = 4$  para todos os problemas. Para o espaço  $\mathbb{R}^S$ , sugerimos  $s = 10$  coordenadas. Desta forma, em todos os problemas propostos o espaço de decisão  $\mathbb{R}^N$  tem  $(M - 1)q + 14$  coordenadas utilizando os valores de  $q$  e  $t$  sugeridos.

#### 4.5.4.1 pd-MOP-01

Este problema apresenta uma Fronteira Pareto com regiões concavas e convexas, que são simétricas à hiper-diagonal  $\mathbf{d} = (1, 1, \dots, 1)$  do primeiro ortante do espaço dos objetivos. Sua avaliação é feita da seguinte forma:

1. Seja  $\mathbf{x}$  um vetor do espaço de decisão. Inicialmente avalie a função auxiliar  $T(\mathbf{x})$  definida pela equação (4.13), cuja imagem está no primeiro ortante do espaço dos objetivos  $\mathbb{R}_M$  e calcule o valor de  $\phi(\mathbf{x})$  por meio das equações (4.15) e (4.16).
2. Calcule o valor de  $F_d(\mathbf{x})$  por meio da expressão

$$F_d(\mathbf{x}) = \frac{\phi(\mathbf{x})^\alpha}{2} + g(\mathbf{x}) + 0.5 \quad (4.30)$$

sendo  $\alpha$  um parâmetro com valor sugerido de  $\alpha = 5$  e  $g(\mathbf{x})$  a função deceptiva definida pela equação (4.17). Para a versão robusta deste problema use a função  $g(\mathbf{x})$  robusta definida pela equação (4.19).

3. Caso se opte por objetivos não dissimilares, defina a função  $F_p(\mathbf{x})$  como  $F_p(\mathbf{x}) = T(\mathbf{x})$  sendo  $T(\mathbf{x})$  descrita pela equação (4.13). Caso se opte por objetivos dissimilares, defina a função  $F(\mathbf{x}) = D(\mathbf{x})$  sendo  $D(\mathbf{x})$  definida pela equação (4.14).
4. Opcionalmente acrescente alguma das restrições definidas na seção 4.5.3.

O conjunto Pareto ótimo desta função é  $0 \leq x_i \leq 1$  para  $x_i \in \mathbb{R}^R$  e  $x_j = 0.5$  para  $x_j \in \mathbb{R}^S$ . A Fronteira Pareto deste problema é apresentada nas figuras 23a e 23b utilizando  $\mathbf{d} = (1, 1)$  para o problema com dois objetivos e  $\mathbf{d} = (1, 1, 1)$  para o problema com três objetivos.

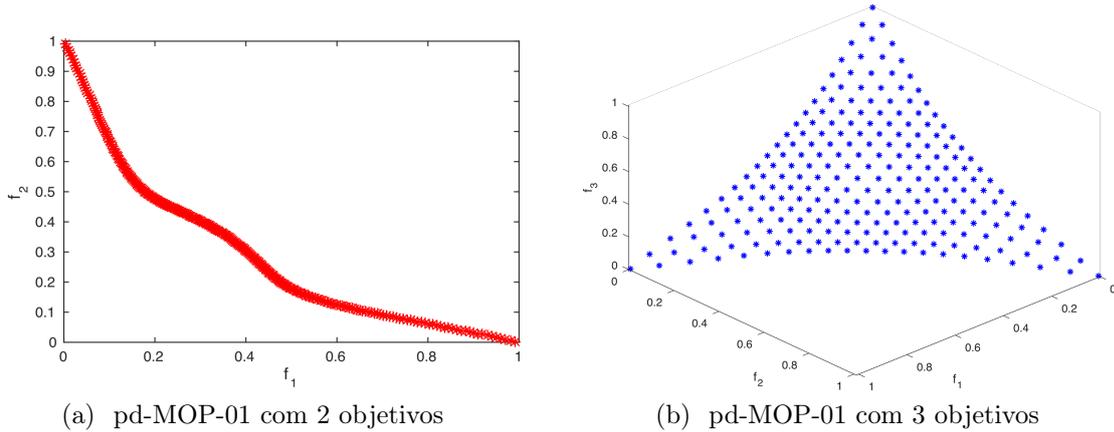


Figura 23 – Fronteira Pareto do problema pd-MOP-01 não dissimilar com 2 e 3 objetivos.

#### 4.5.4.2 pd-MOP-02

Este problema permite a modificação da concavidade da Fronteira Pareto por meio da definição do parâmetro  $p$  que define a norma (ou quasi-norma) utilizada na função  $h(\mathbf{x}) = \|T(\mathbf{x})\|_p$  presente na equação (4.12). Se  $0 < p < 1$  a Fronteira Pareto é convexa. Se  $p = 1$  a Fronteira Pareto é um hiperplano e se  $p > 1$  o problema é concavo. Para problemas com dois ou três objetivos os valores de  $p$  iguais a 0.5, 1, 2 e 3 são sugeridos.

Em problemas com mais de três objetivos os objetivos se tornam desbalanceados. Por exemplo, no espaço com  $M$  dimensões um vetor da base canônica  $\mathbf{e}_i = (0, \dots, 0, 1, 0, \dots, 0)$  tem  $p$  norma igual a 1 para qualquer valor de  $p > 0$ . Para vetores localizados no bordo do primeiro ortante do espaço  $\mathbb{R}^M$  um vetor com  $p$ -norma constante igual a 1 o valor de suas coordenadas decresce. No caso extremo, um vetor  $\mathbf{v}$  paralelo à hiper-diagonal  $\mathbf{d} = (1, \dots, 1)$  com  $p$ -norma igual a 1 tem as seguintes coordenadas:

$$\mathbf{v} = \frac{\mathbf{d}}{\|\mathbf{d}\|_p} \quad (4.31)$$

$$= \left( \frac{1}{\sqrt[p]{M}}, \dots, \frac{1}{\sqrt[p]{M}} \right) \quad (4.32)$$

Para um valor de  $p$  fixo,  $\frac{1}{\sqrt[p]{M}}$  decresce muito rapidamente. A figura 24 apresenta a evolução dos valores de  $\frac{1}{\sqrt[p]{M}}$  em função da dimensão  $M$  do espaço. O decaimento deste valor é mais intenso para valores de  $p > 3$ , que ficam próximos de zero em problemas de alta dimensionalidade. Para que  $\frac{1}{\sqrt[p]{M}}$  não assuma valores extremos, basta determinarmos um valor ideal para  $p$  nestas situações. Por exemplo, para que  $\frac{1}{\sqrt[p]{M}} > \frac{1}{2}$ , basta que  $p > \frac{\ln(M)}{\ln(2)}$ . Desta forma para problemas com mais de 3 objetivos sugerimos  $p = \left\lceil \frac{\ln(M)}{\ln(2)} \right\rceil$ .

A avaliação do problema **pd-MOP-02** é feita da seguinte maneira:

1. Seja  $\mathbf{x}$  um vetor do espaço de decisão. Caso se opte por objetivos não dissimilares,

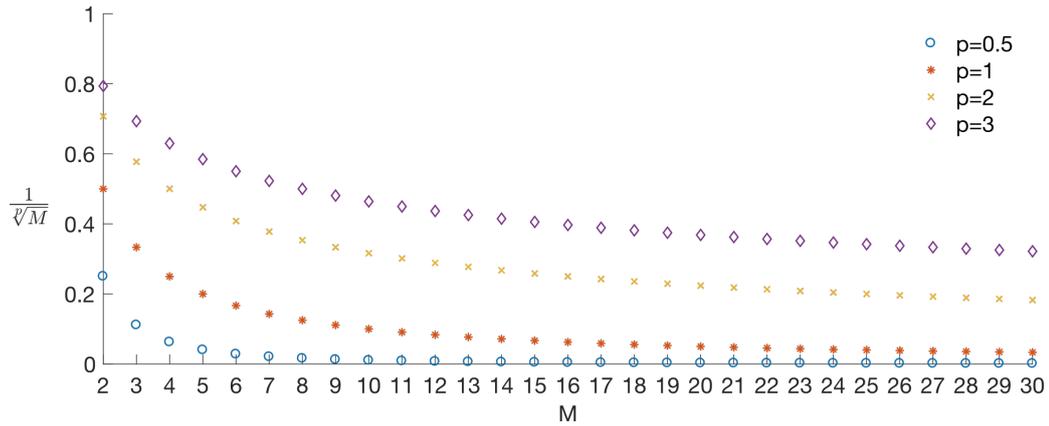


Figura 24 – Valor de  $\frac{1}{\psi_M}$  em espaços  $M$ -dimensionais

defina a função  $F_p(\mathbf{x})$  como  $F_p(\mathbf{x}) = \frac{T(\mathbf{x})}{h(\mathbf{x})}$  como descrita pela equação (4.13). Caso se opte por objetivos dissimilares, use a função  $D(\mathbf{x})$  definida pela equação (4.14).

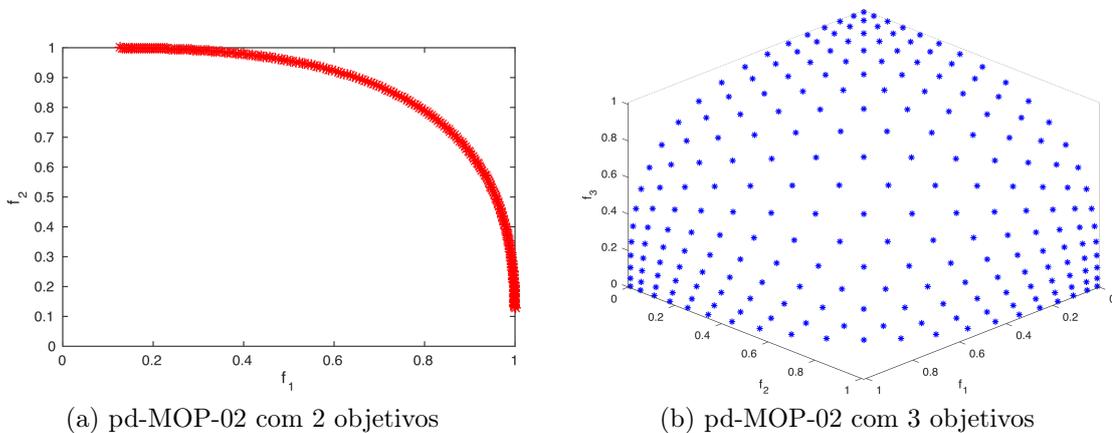
2. Calcule o valor de  $F_d(\mathbf{x})$  por meio da expressão

$$F_d(\mathbf{x}) = g(\mathbf{x}) + 1 \tag{4.33}$$

sendo  $g(\mathbf{x})$  a função deceptiva definida pela equação (4.17). Para a versão robusta deste problema use a função  $g(\mathbf{x})$  robusta definida pela equação (4.19).

3. Opcionalmente acrescente alguma das restrições definidas na seção 4.5.3.

O conjunto Pareto ótimo desta função é  $0 \leq x_i \leq 1$  para  $x_i \in \mathbb{R}^R$  e  $x_j = 0.5$  para  $x_j \in \mathbb{R}^S$ . A Fronteira Pareto deste problema com  $p = 3$  é apresentada nas figuras 25a e 25b.



(a) pd-MOP-02 com 2 objetivos

(b) pd-MOP-02 com 3 objetivos

Figura 25 – Fronteira Pareto do problema pd-MOP-02 não dissimilar com 2 e 3 objetivos com  $p = 3$ .

#### 4.5.4.3 pd-MOP-03

Este problema apresenta uma Fronteira Pareto desconexa, simétrica em relação à hiper-diagonal  $\mathbf{d} = (1, \dots, 1)$  do primeiro ortante do espaço dos objetivos. Sua avaliação é feita da seguinte forma:

1. Seja  $\mathbf{x}$  um vetor do espaço de decisão. Inicialmente avalie a função auxiliar  $T(\mathbf{x})$  definida pela equação (4.13), cuja imagem está no primeiro ortante do espaço dos objetivos  $\mathbb{R}_M$  e calcule o valor de  $\phi(\mathbf{x})$  por meio das equações (4.15) e (4.16).
2. Calcule o valor de  $F_d(\mathbf{x})$  por meio da expressão

$$F_d(\mathbf{x}) = \frac{\cos(3\pi\phi)^2}{10} + g(\mathbf{x}) + 1 \quad (4.34)$$

sendo  $g(\mathbf{x})$  a função deceptiva definida pela equação (4.17). Para a versão robusta deste problema use a função  $g(\mathbf{x})$  robusta definida pela equação (4.19).

3. Caso se opte por objetivos não dissimilares, defina a função  $F_p(\mathbf{x})$  como  $F_p(\mathbf{x}) = T(\mathbf{x})$  sendo  $T(\mathbf{x})$  descrita pela equação (4.13). Caso se opte por objetivos dissimilares, defina a função  $F(\mathbf{x}) = D(\mathbf{x})$  sendo  $D(\mathbf{x})$  definido pela equação (4.14).
4. Opcionalmente acrescente alguma das restrições definidas na seção 4.5.3.

O conjunto Pareto ótimo desta função é  $0 \leq x_i \leq 1$  para  $x_i \in \mathbb{R}^R$  e  $x_j = 0.5$  para  $x_j \in \mathbb{R}^S$ . A Fronteira Pareto deste problema é apresentada nas figuras 26a e 26b.

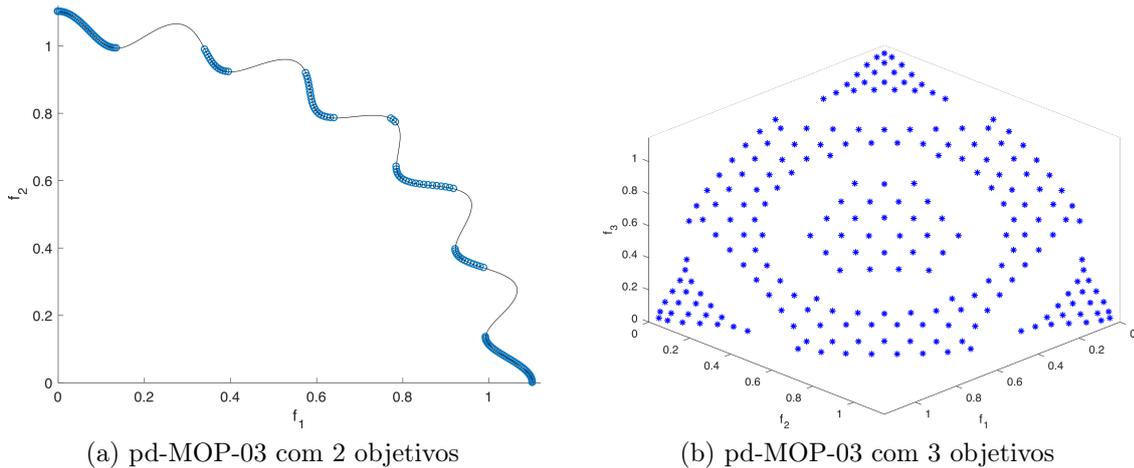


Figura 26 – Fronteira Pareto do problema pd-MOP-03 não dissimilar com 2 e 3 objetivos.

## 4.6 Métrica de Distância Radial

A qualidade de uma solução obtida por meio de um algoritmo evolutivo pode ser estimada pela sua distribuição e convergência em relação a Fronteira Pareto do problema.

Uma métrica de convergência faz uma estimativa da distância entre as soluções obtidas e a Fronteira Pareto. A rigor, a distância  $D(\mathbf{x}, \mathcal{S})$  entre um ponto  $\mathbf{x}$  e uma superfície  $\mathcal{S}$  (ou algum conjunto de pontos  $\mathcal{S}$ ) é definida pela equação

$$D(\mathbf{x}, \mathcal{S}) = \inf \{D(\mathbf{x}, \mathbf{y}), \mathbf{y} \in \mathcal{S}\} \quad (4.35)$$

sendo  $D(\mathbf{x}, \mathbf{y})$  a distância entre os pontos  $\mathbf{x}$  e  $\mathbf{y}$ . Como existem infinitos pontos na Fronteira Pareto, é impossível calcular a distância entre uma solução particular  $\mathbf{x}$  e este conjunto. Para contornar este problema, as métricas de convergência (como por exemplo a *Generational Distance* (GD) [148] ou a *Inverted Generational Distance* (IGD) [23] utilizam uma amostra de pontos igualmente distribuídos contidos na Fronteira Pareto. Entretanto, este procedimento faz com que o valor obtido por estas métricas seja extremamente dependente da qualidade e da quantidade de pontos desta amostra, principalmente em problemas com muitos objetivos [55]. Uma forma de se evitar a dependência causada pela diferença entre as amostras é criar uma métrica de convergência que dependa apenas da solução obtida e da localização da Fronteira Pareto.

Seja  $\mathbf{x} = (\mathbf{x}_p, \mathbf{x}_d)$  uma solução de um problema separável em parâmetros de posição e distância, cuja solução ótima  $\mathbf{x}^* = (\mathbf{x}_p^*, \mathbf{x}_d^*)$  é conhecida e considere o ponto  $\mathbf{y} = (\mathbf{x}_p, \mathbf{x}_d^*)$ . Então os vetores definidos por  $F(\mathbf{x})$  e  $F(\mathbf{y})$  no espaço de decisão são paralelos uma vez que  $\mathbf{x}$  e  $\mathbf{y}$  possuem a mesma variável de posição  $\mathbf{x}_p$ . Além disso, o ponto  $F(\mathbf{y})$  está contido na Fronteira Pareto pois  $\mathbf{x}_d^*$  é uma valor ótimo. Desta forma  $|F(\mathbf{x}) - F(\mathbf{y})|$  é uma medida de distância entre a solução  $F(\mathbf{x})$  e a Fronteira Pareto. Note que  $|F(\mathbf{x}) - F(\mathbf{y})| \neq D(F(\mathbf{x}), PF)$ , mas este valor não depende de uma amostra de pontos neste conjunto.

Para cada solução  $\mathbf{x}^i = (\mathbf{x}_{p_i}, \mathbf{x}_{d_i})$  de um conjunto de soluções  $\{\mathbf{x}^1, \dots, \mathbf{x}^k\}$ , considere os pontos  $F(\mathbf{x}^i)$  e  $F(\mathbf{y}^i)$  do espaço de decisão. A Métrica de distância Radial (*Radial Distance Metric* - RDM) é definida como

$$RDM = \frac{1}{k} \sum_{j=1}^k \|F(\mathbf{x}_j) - F(\mathbf{y}_j)\| \quad (4.36)$$

Pelas considerações apresentadas anteriormente a métrica RDM não depende de uma amostra de pontos na Fronteira Pareto e se aplica a problemas de teste cujas variáveis sejam separáveis em parâmetros de posição e distância, cuja solução ótima seja conhecida.

## 4.7 Considerações finais

Funções de *benchmark* juntamente com as métricas de avaliação de desempenho são um importante instrumento de validação de algoritmos de otimização. As funções avaliadas neste capítulo apresentam muitas virtudes, mas também têm limitações. O novo

conjunto de funções de teste proposto procura manter as qualidades dos problemas de teste apresentados e superar algumas de suas virtudes. Todos os problemas propostos são de fácil implementação e interpretação, têm soluções conhecidas e não apresentam limitação quanto ao número de objetivos. Também possuem uma boa variedade de opções de formato e topologias da Fronteira Pareto além de apresentarem um significativo desafio aos algoritmos de otimização de encontrar um conjunto de soluções de boa qualidade.

O principal destaque deste novo conjunto de funções de teste é a possibilidade de se criar problemas de otimização robusta com diversas características, que constitui uma grande lacuna na literatura especializada. Outra característica pouco explorada nas propostas de problemas de otimização são as restrições de desigualdade e de igualdade, que nas funções apresentadas têm uma formulação clara e de fácil interpretação.

A métrica de desempenho proposta apresenta importantes características, sendo o seu destaque a independência de uma amostra de pontos da Fronteira Pareto e terá um papel importante na avaliação de soluções localizadas em regiões específicas do Espaço dos Objetivos, tema que será tratado no capítulo 7.

## 5 Visualização de dados em espaços de dimensão elevada

A visualização de dados em espaços de alta dimensionalidade é um grande desafio pois de maneira geral dispomos apenas de dispositivos bidimensionais para sua representação, tais como uma folha de papel ou uma tela de dispositivo eletrônico. Um exemplo da complexidade da representação de dados variados é a figura 27 criada pelo engenheiro civil Charles Joseph Minard, que foi superintendente da *École Nationale des Ponts et Chaussées*. Este gráfico <sup>1</sup>, chamado de *Carte figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812-1813*, representa a desastrosa campanha do exército de Napoleão na Rússia.

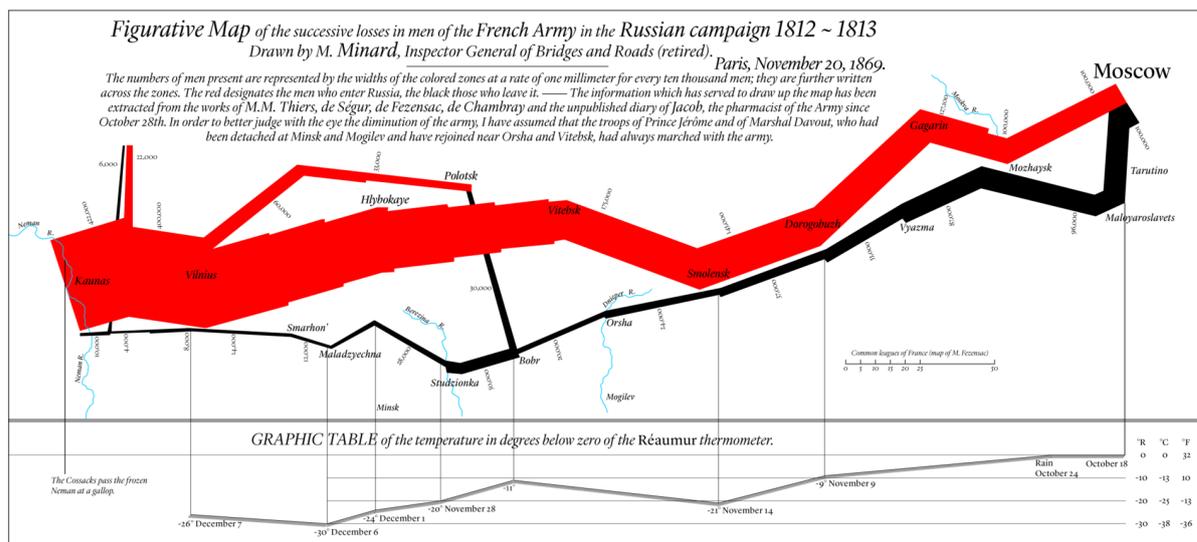


Figura 27 – Carte figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812-1813

Nesta figura estão presentes seis variáveis:

1. Tamanho do exército, representado pela espessura das linhas vermelhas e pretas;
2. Distância percorrida;
3. Temperatura em cada seção (abaixo do gráfico);
4. Localização espacial do exército, com a latitude e longitude;
5. Localização temporal do posicionamento das tropas;

<sup>1</sup> O original está disponível no endereço [https://patrimoine.enpc.fr/document/ENPC01\\_Fol1\\_10975?image=54#bibnum](https://patrimoine.enpc.fr/document/ENPC01_Fol1_10975?image=54#bibnum)

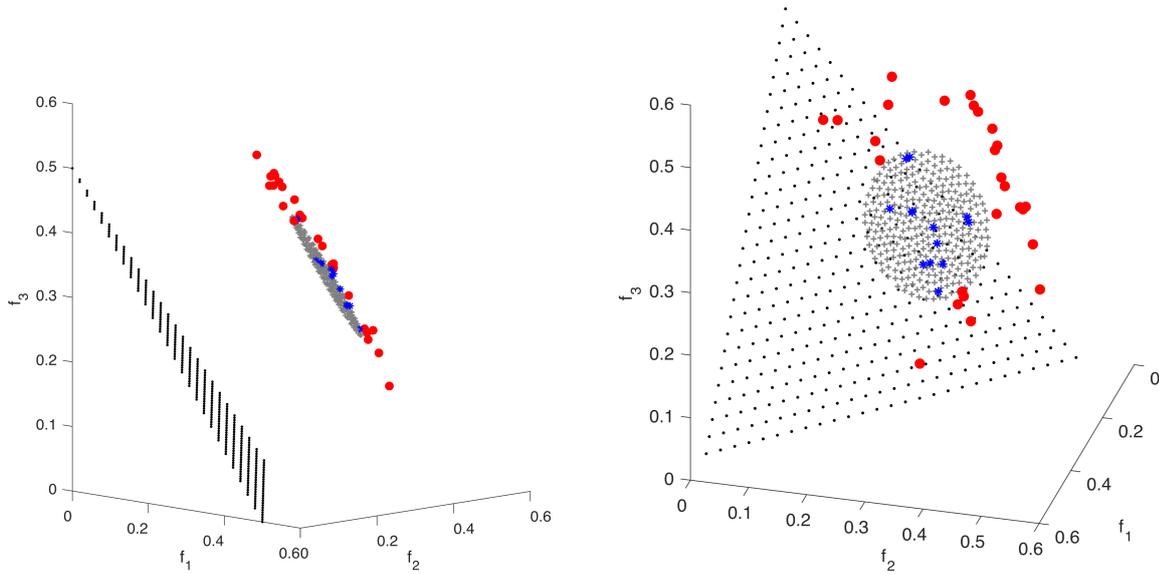


Figura 28 – Duas representações para o mesmo conjunto de pontos

6. Direção da marcha (vermelho na ida e preto no retorno).

De maneira geral, para representar soluções de problemas de otimização multiobjetivo nos espaços  $\mathbb{R}^2$  e  $\mathbb{R}^3$  é utilizado um gráfico de dispersão de pontos (*scatter plot*). Entretanto, em algumas situações, mesmo esta representação básica pode apresentar problemas. A figura 28 apresenta duas representações gráficas de quatro conjuntos de pontos no espaço no primeiro octante do  $\mathbb{R}^3$ , utilizando diferentes ângulos de visão. Na figura da esquerda percebe-se com clareza a formação de dois agrupamentos de pontos, mas sem muita informação sobre sua dispersão. Na figura da direita a dispersão dos pontos é nítida, mas se perde a informação da formação dos agrupamentos.

Em otimização multiobjetivo o que se espera de uma boa representação gráfica de uma solução em particular é poder verificar a convergência e a dispersão da mesma, bem como a harmonia e o conflito entre diferentes objetivos. Neste sentido muitos pesquisadores dedicam seu trabalho a este tema, principalmente em problemas com mais do que três objetivos. O mais popular método de representação de soluções em espaços de alta dimensionalidade é o método das coordenadas paralelas [102]. Li et al. [85] faz uma interessante discussão sobre como interpretar os resultados apresentados por este método.

O método das árvores de agregação (*Aggregation Trees*) [39, 40] é baseado na agregação sequencial dos objetivos que são representados em uma árvore, considerando o conflito entre pares (ou grupos) de objetivos.

Um outro método aplicável a problemas com elevada dimensionalidade é o RadViz (*Radial Coordinate Visualization*) [152, 153]. Neste método, inspirado no modelo mecânico

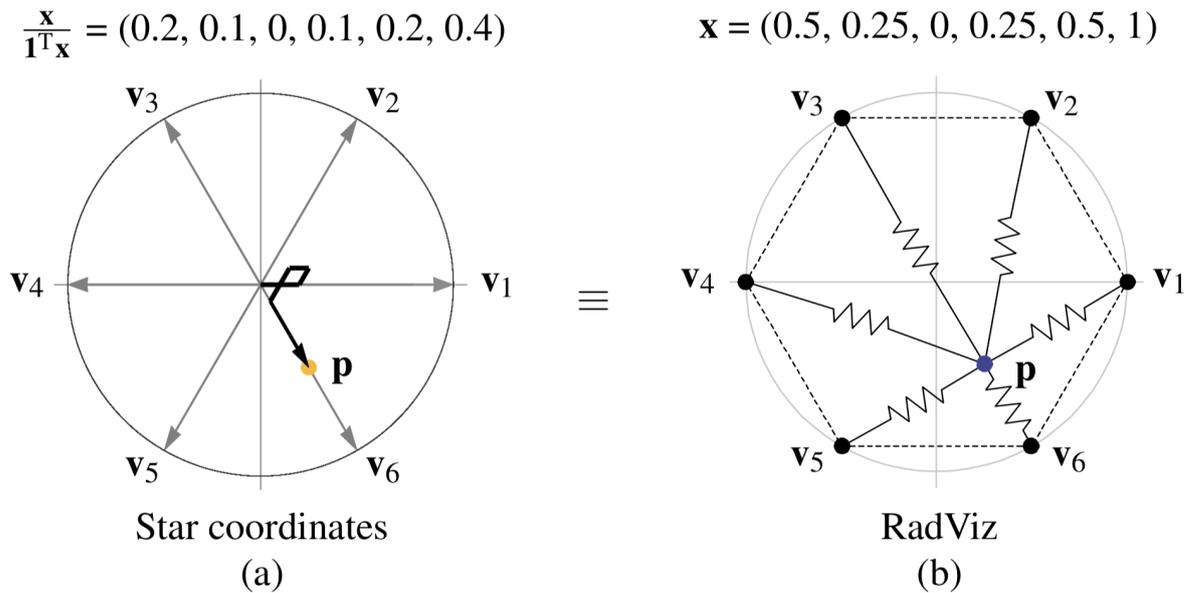


Figura 29 – Representação de um mesmo ponto no sistema de Coordenadas Estelares e no método RadViz. Figura retirada de Rubio-Sanchez et al. [125].

de molas, os  $M$  objetivos são dispostos como vértices de um polígono regular de  $M$  lados e são denominados de âncoras. Os objetivos são normalizados no intervalo  $[0, 1]$  e seus valores são associados ao coeficiente de elasticidade de uma mola dado pela lei de Hooke. As molas estão conectadas nas âncoras e em um ponto que representa o ponto que se quer representar, posicionando este no interior do polígono. Opcionalmente este polígono pode ser não regular, de acordo com a matriz de covariância entre as soluções.

No método das coordenadas estelares (*star coordinates*) [70, 163] os  $M$  objetivos são representados em um sistema de coordenadas composto de  $M$  vetores unitários  $\{C_1, \dots, C_M\}$ , igualmente espaçados e dispostos em um círculo. Uma solução  $\mathbf{y}$  é normalizada de modo que sua norma  $\ell_1$  seja constante e igual a um, ou seja, o vetor  $\mathbf{y}$  é transformado no vetor  $\mathbf{x} = \frac{\mathbf{y}}{|\mathbf{y}|_1}$  de modo que  $|\mathbf{x}|_1 = 1$ . As coordenadas do vetor normalizado  $\mathbf{x} = (x_1, \dots, x_M)$  são utilizadas como pesos na combinação convexa de vetores  $C = \sum_{i=1}^M x_i C_i$ . O vetor  $C$  resultante desta soma representa a posição final do ponto  $\mathbf{y}$  neste método de visualização. Rubio-Sanchez et al. [125] mostra que este método é equivalente ao RadViz e apresenta representações similares. Entretanto o método das coordenadas estelares é superior em alguns aspectos: tem uma representação matemática mais simples e apresenta menos distorções. A figura 29 ilustra um mesmo ponto do espaço  $\mathbb{R}^6$  representado nos dois sistemas.

Os Mapas de Calor (*Heatmap*) [118, 57] utilizam gradientes de cores para diferenciar características de pontos, que são usualmente combinadas com outras técnicas de representação gráfica de informação. Outras técnicas de representação gráfica são os Mapas Auto-Organizados (*Self-Organized Map*) [73, 37], Gráficos Radar (*Radar Graph*) [39, 58],

ISOMap [149] e Sammon mapping [56, 147].

He and Yen [55] apresentam um método de visualização onde pontos de um espaço de alta dimensionalidade são projetados no plano polar bidimensional, preservando importantes características tais como Dominância Pareto (em alguns casos), forma e localização da Fronteira Pareto e a distribuição dos indivíduos.

Koochaksaraei et al. [76] apresentam dois métodos de visualização de dados em espaços de alta dimensionalidade: O diagrama de Cordas (*Chord Diagram*) e o Mapeamento Angular (*Angular Mapping*), ambos seguindo um *design* circular. Estas técnicas fazem o mapeamento de pontos localizados em um espaço de dimensão  $M$  em um espaço de dimensão dois.

No Diagrama de Cordas os objetivos são normalizados no intervalo  $[0, 1]$  e são representados por meio de arcos em um círculo. Cada ponto é representado por uma curva de Bézier, conectando cada coordenada em cada arco. Uma importante característica presente em problemas de otimização multiobjetivo é a harmonia/conflito entre diferentes objetivos. A figura 30 apresenta um diagrama de cordas para pontos no espaço  $\mathbb{R}^{10}$ . No primeiro objetivo, rotulado como Obj0, foi destacado um conjunto de pontos com valores nominais localizados na primeira metade do intervalo (linhas verdes). Nesta figura é possível observar que estes valores estão associados a valores na primeira metade dos objetivos 2,3,4 e 5, caracterizando a harmonia entre estes objetivos. Este padrão é facilmente detectável pela existência de linhas que se cruzam no interior do diagrama. Já a conexão com os objetivos 6,7,8 e 9 acontece na segunda metade destes objetivos, caracterizando o conflito. Este padrão é caracterizado pela presença de linhas paralelas conectando estes pontos.

O mapeamento angular faz a representação de um ponto no espaço  $M$ -dimensional considerando sua proximidade (angular) com os vetores  $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_M$  da base canônica. Para isto são utilizadas três características de um ponto  $\mathbf{y} = (y_1, \dots, y_M)$ : sua norma euclidiana  $\rho$ , o menor ângulo  $\theta$  entre o vetor  $\mathbf{y}$  e os vetores da base canônica e o índice  $j$  que indica em qual objetivo o menor ângulo foi observado. Estes valores são calculados de acordo com as fórmulas apresentadas nas equações (5.1), sendo  $\alpha_i$  o ângulo entre o vetor  $\mathbf{y}$  e o vetor da base canônica  $\mathbf{e}_i$ .

$$\begin{aligned}\alpha_i &= \arccos\left(\frac{y_i}{\rho}\right), \quad i = 1, \dots, M \\ \theta &= \alpha_j = \min_i \alpha_i \\ j &= \arg \min_i \alpha_i\end{aligned}\tag{5.1}$$

O valor máximo do ângulo  $\theta$  é  $\frac{1}{\sqrt{M}}$ , que corresponde ao ângulo entre qualquer vetor  $\mathbf{e}_i$  da base canônica e a hiper-diagonal  $\mathbf{d} = (1, \dots, 1)$  do primeiro ortante do espaço  $\mathbb{R}^M$ . O índice  $j$  faz uma classificação dos pontos do espaço  $\mathbb{R}^M$  de acordo com sua proximidade

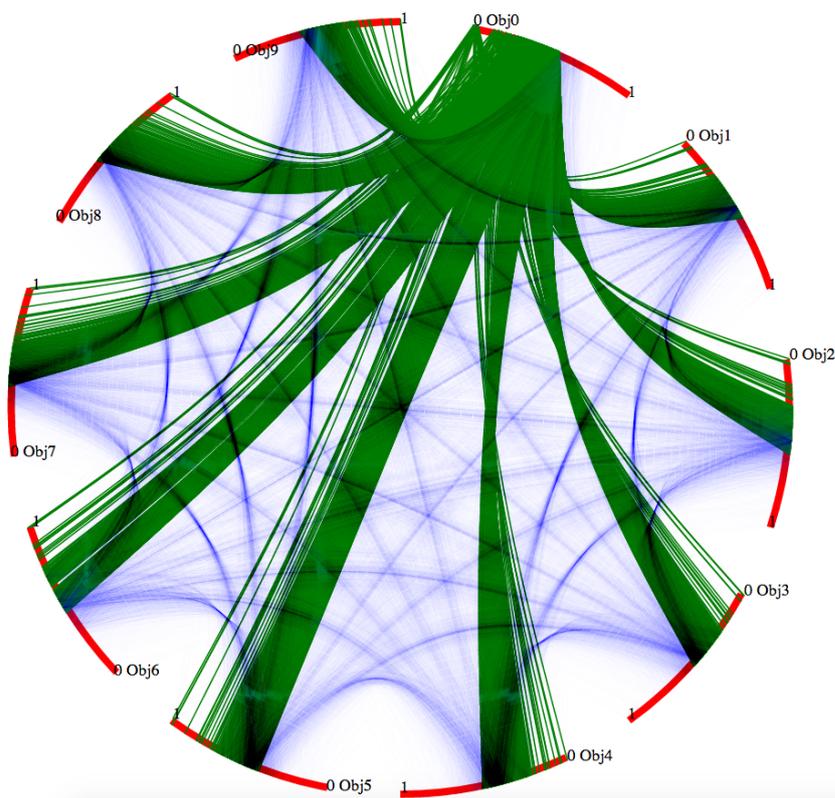
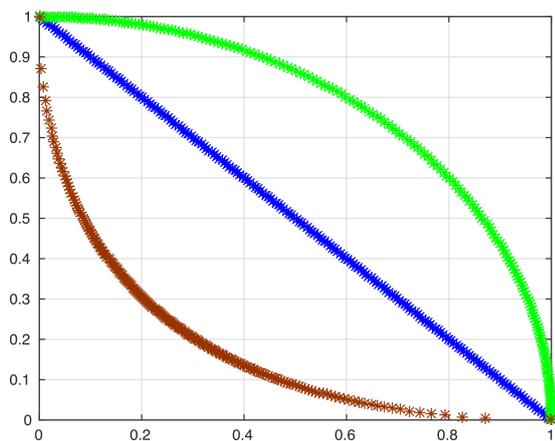


Figura 30 – Diagrama de cordas de um conjunto de pontos no espaço  $\mathbb{R}^{10}$ . Os objetivos estão indexados de 0 a 9

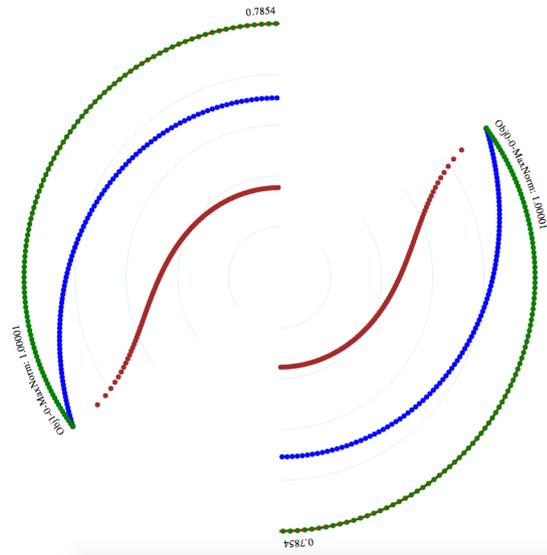
com os eixos da base canônica  $\mathbf{e}_i$ , mas esta classificação apresenta algumas limitações. Por exemplo, na hiper-diagonal do primeiro ortante  $\mathbf{d} = (1, \dots, 1)$  o ângulo entre os eixos é o mesmo. Em situações como esta, o índice  $j$  é escolhido aleatoriamente entre as opções disponíveis. Além disso, todo vetor contido na extremidade da base de um cone de altura  $h$  fixa e cujo eixo é um dos vetores  $\mathbf{e}_i$  tem o mesmo valor de  $[\rho, \theta, j]$  o que faz com que mais de um ponto no espaço  $\mathbb{R}^M$  tenha a mesma representação.

A figura 31 apresenta uma comparação entre a representação de pontos utilizando o diagrama de dispersão de pontos e o mapeamento angular nos espaços  $\mathbb{R}^2$  e  $\mathbb{R}^3$ . Cada gráfico contém três conjuntos de pontos igualmente espaçados em superfícies de  $p$ -norma constante, com  $p$  iguais a 0.5, 1 e 2, formando um conjunto de pontos convexo, plano e concavo respectivamente.

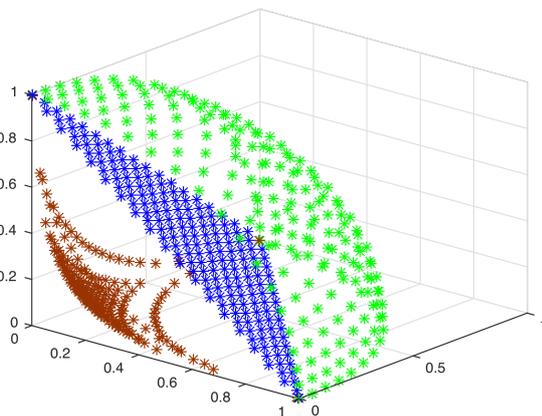
Observe que a representação de pontos utilizando o mapeamento angular é capaz de preservar a posição relativa de cada um destes conjuntos de pontos com características próprias. Por exemplo, os pontos contidos na superfície de norma euclidiana constante ( $p$  igual a dois) são mapeados na borda do gráfico de mapeamento angular. Os pontos próximos à hiper-diagonal  $\mathbf{d} = (1, \dots, 1)$  em cada conjunto de pontos estão mais afastados tanto na representação do gráfico de dispersão quanto no mapeamento angular. Já os pontos próximos aos vetores da base canônica  $\mathbf{e}_i$  estão próximos entre si nas duas representações. Maiores detalhes sobre o diagrama de cordas e o mapeamento angular podem ser



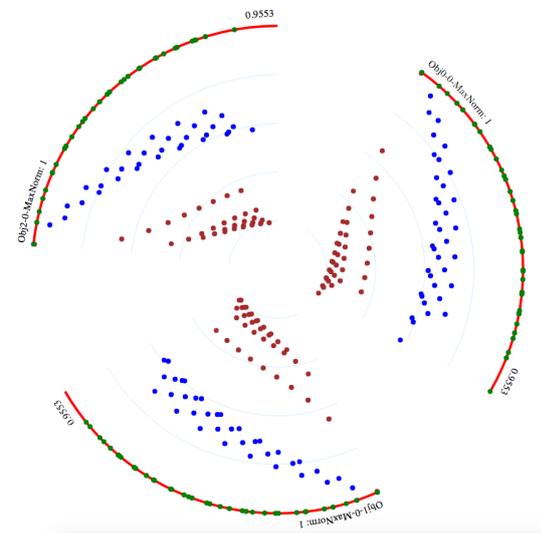
(a) Pontos em  $\mathbb{R}^2$  representados por meio do diagrama de dispersão de pontos.



(b) Pontos da figura 31a representados por meio do Mapeamento Angular.



(c) Pontos em  $\mathbb{R}^3$  representados por meio do diagrama de dispersão de pontos



(d) Pontos da figura 31c representados por meio do Mapeamento Angular.

Figura 31 – Três conjuntos de pontos representados utilizando o gráfico de dispersão de pontos e o mapeamento angular em espaços com duas e três dimensões

encontradas em [75, 74].

Em todos os métodos de representação de pontos de espaços com dimensão maior do que três apresentados nesta seção está presente alguma perda de informação. Nos métodos de coordenadas paralelas, árvores de agregação e diagrama de cordas ocorre a perda de informação visual uma vez que se percebe apenas a relação entre as coordenadas dos pontos e/ou sua proximidade. Além disso, nestes métodos, um ponto é representado por meio de segmentos de retas ou de curvas o que não está de acordo com nossa percepção do que seja um ponto. Nos métodos RADviz, coordenadas estelares e mapeamento angular um ponto é representado de forma usual, ou seja, um ponto mesmo. Nos métodos RAD-viz e coordenadas estelares o primeiro passo é a normalização do ponto  $\mathbf{y}$  do espaço  $\mathbb{R}^M$ , fazendo  $\mathbf{x} = \frac{\mathbf{y}}{|\mathbf{y}|_p}$  sendo  $p = 2$  no RAD-viz e  $p = 1$  nas coordenadas estelares. Desta forma, todo ponto  $\mathbf{w}$  da forma  $\mathbf{w} = \lambda\mathbf{y}$  com  $\lambda > 0$  é representado pelo mesmo ponto  $\mathbf{x}$  após a normalização. Desta forma, estes métodos não são capazes de preservar a posição relativa dos pontos em relação à origem, mas esta limitação está presente no mapeamento angular como ilustrado nas figuras 31a e 31b.

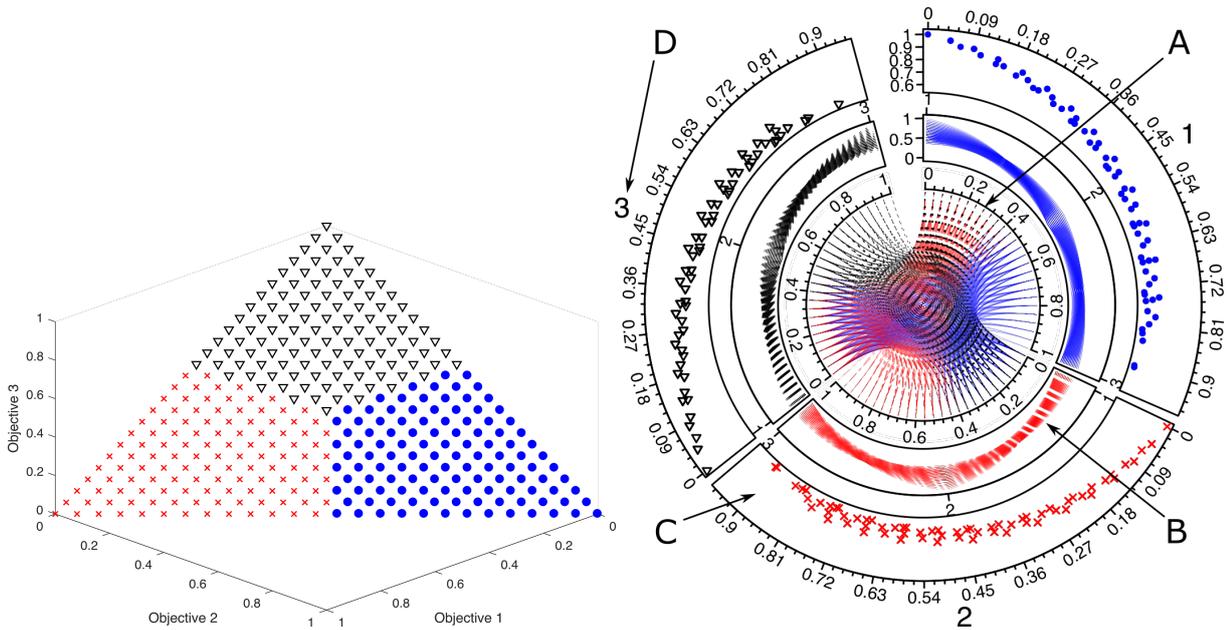
## 5.1 CAP-vis

Uma vez que a perda de informação é inevitável, esta deve ser compensada. Este trabalho apresenta um novo método de visualização de pontos em espaços de elevada dimensionalidade em uma figura bidimensional, chamada de CAP-vis [101]. Este método reúne o diagrama de Cordas, o mapeamento Angular e o método das coordenadas Paralelas em uma única figura, utilizando o *design* circular. O *design* circular é uma forma eficiente e popular de representação de dados em espaços de alta dimensionalidade [77]<sup>2</sup>. Por utilizar três métodos distintos de visualização de dados, a perda de informação que ocorre em um método pode ser compensada pela informação contida em outro método.

O *layout* do gráfico CAP-Vis é composto por setores e trilhas. Os setores correspondem à classificação dos pontos por sua distância angular aos eixos do espaço  $\mathbb{R}^M$  e cada trilha corresponde a um método de visualização. Estes elementos são apresentados na figura 32, onde os pontos associados a cada setor estão destacados com cores diferentes. Na trilha A é representado o Diagrama de Cordas, na trilha B está presente o gráfico de Coordenadas Paralelas e na trilha C se localiza o Mapeamento angular. A classificação dos pontos em diferentes setores é feita por meio das equações (5.1). Nas trilhas B e C estão presentes apenas os pontos associados ao seu respectivo setor enquanto que na trilha A estão representados todos os pontos.

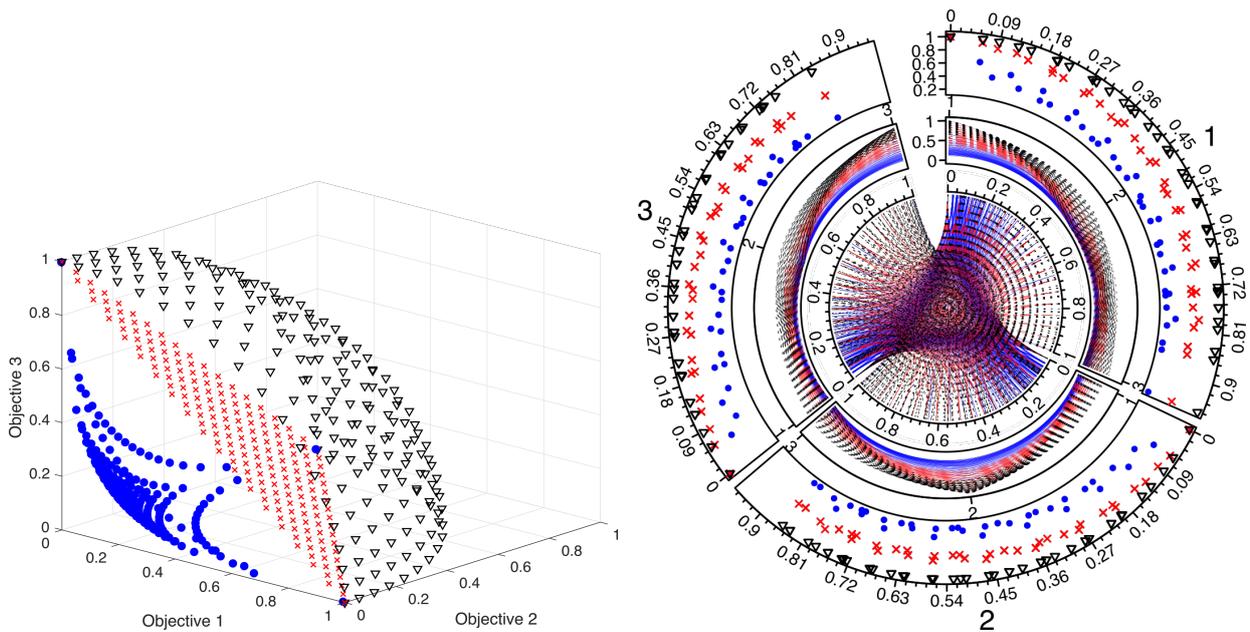
O Mapeamento Angular preserva a posição relativa dos pontos e esta característica também está presente no CAP-Vis. A figura 33 apresenta três conjuntos de pontos

<sup>2</sup> Um rico exemplo desta técnica pode ser encontrada no site <http://circos.ca/>



(a) Pontos em um plano utilizando o diagrama de dispersão de pontos. (b) Pontos em um plano utilizando o CAP-Vis.

Figura 32 – O mesmo conjunto de pontos representados no diagrama de dispersão de pontos no CAP-Vis e sua classificação pela aproximação angular com os vetores da base canônica e seus elementos gráficos: A: Diagrama de cordas; B: Coordenadas Paralelas; C: Mapeamento angular; D: Índice da região representada.



(a) Representação utilizando o diagrama de dispersão de pontos. (b) Representação utilizando o CAP-Vis.

Figura 33 – Pontos em superfícies de norma (quasi norma) constante no espaço  $\mathbb{R}^3$

localizados em superfícies de norma (quasi-norma) constante, representados pelo diagrama de dispersão de pontos (figura 33a) e pela ferramenta CAP-Vis (figura 33b). Como os pontos localizados na superfície de quasi-norma constante ( $p = 0.5$ ) estão localizados muito próximos da origem do sistema de coordenadas do diagrama de dispersão de pontos, suas coordenadas são menores do que as coordenadas dos pontos localizados nas demais superfícies. Este fenômeno pode ser observado no gráfico de Diagrama de Cordas (trilha A) e no gráfico de coordenadas Paralelas (trilha B) pela maior concentração das linhas azuis em coordenadas de menor valor. Entretanto, este fenômeno é muito mais fácil de ser visualizado no Mapeamento Angular (trilha C).

O CAP-Vis foi implementado na linguagem R, utilizando a biblioteca *circlize* [50]. Uma cópia está disponibilizada no site do MINDS (<http://minds.eng.ufmg.br/>) e no github (<https://github.com/ivanreinaldo/CAP-Vis>).

## 5.2 Aplicações

### 5.2.1 Visualização de soluções

A figura 34 faz uma comparação entre a visualização de soluções de um problema de otimização com muitos objetivos. Foi utilizado o problema pd-MOP-02 com cinco objetivos dissimilares e o algoritmo NSGAIII implementado na plataforma platEMO. Na representação das soluções em coordenadas paralelas, exibida na figura 34b, as soluções obtidas são representadas por linhas azuis enquanto que o limite das variáveis na Fronteira Pareto é representado por linhas vermelhas. Nesta figura pode ser observada a distribuição inadequada das soluções obtidas nos objetivos 3 e 4 mas não é possível perceber o impacto desta distribuição desbalanceada no posicionamento das soluções no espaço dos objetivos. A figura 34a ilustra o mesmo conjunto de soluções representada na ferramenta CAP-vis. Nesta representação é possível perceber o impacto da distribuição inadequada das soluções: existem poucas soluções próximas dos objetivos 2, 3 e 4. Observe que seria previsível alguma degradação na distribuição das soluções na região próxima aos objetivos 3 e 4, mas esta distribuição inadequada também afetou o posicionamento espacial das soluções próximas ao objetivo 2 que também é uma região com poucas soluções apesar de apresentar uma distribuição de soluções similar aos objetivos 1 e 5 como pode ser observado na figura 34b.

Uma outra questão que se observa na figura 34a é a distribuição das soluções próximas ao objetivo 5. Esta região, apesar de ter um bom número de soluções quando comparada com as regiões próximas aos objetivos 2, 3 e 4, concentra suas soluções na região central do espaço dos objetivos como pode ser percebido no mapeamento angular apresentado na trilha C do CAP-vis. Ao contrário das soluções representadas na trilha C do setor 1, que estão bem distribuídas no intervalo de variação do ângulo  $\theta$  (a saber,  $0 \leq \theta \leq 1.1 \approx \arccos\left(\frac{1}{\sqrt{5}}\right)$ ), as soluções representadas nesta parte do gráfico tem o valor de

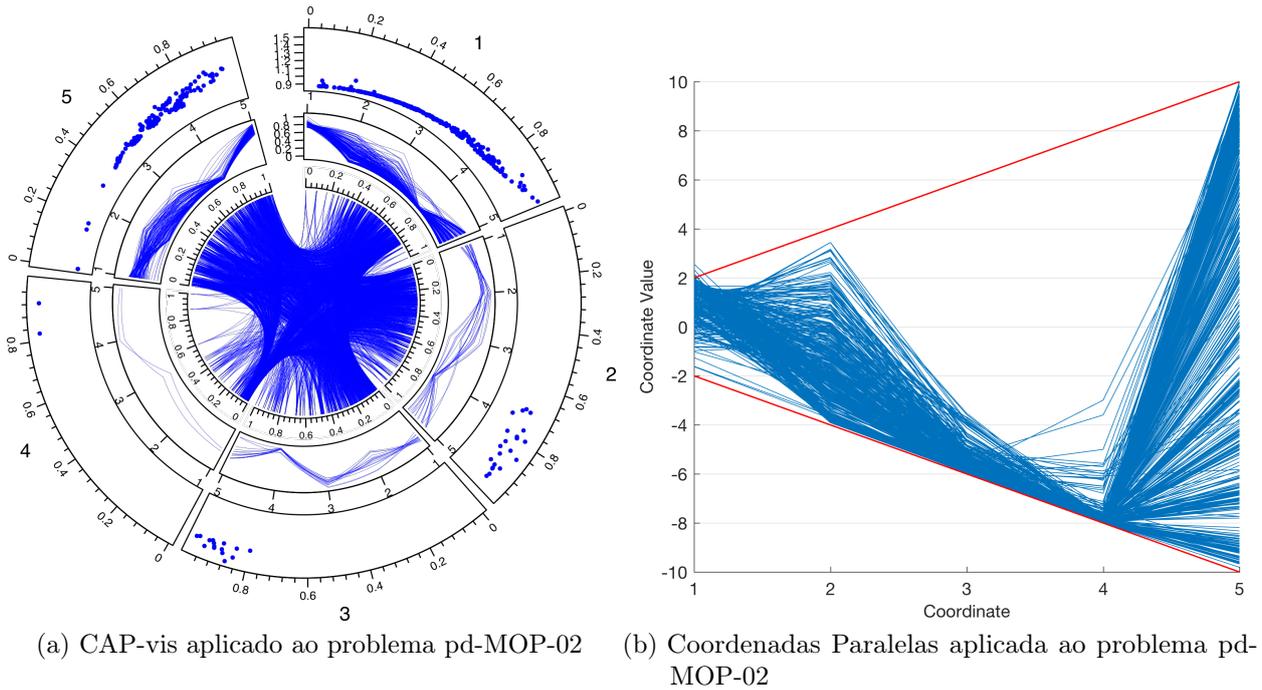


Figura 34 – Soluções obtidas no problema dissimilar pd-MOP-02 com 5 objetivos

$\theta$  concentrado no intervalo  $(0.4, 1)$  com poucas soluções com  $\theta < 0.4$ . Este comportamento pode ser explicado pela distribuição das coordenadas das soluções obtidas no objetivo 1 no gráfico de coordenadas paralelas (ver figura 35a).

A figura 35 apresenta o histograma da distribuição dos valores de cada objetivo deste problema, respeitando os limites de variação das coordenadas de cada solução.

## 5.2.2 Comparação de soluções

Uma outra característica do CAP-vis é sua capacidade de exibir mais de um conjunto de dados. Para ilustrar esta capacidade, considere o problema robusto pd-MOP-02. Este problema tem como solução ótima não robusta  $x_i = 0.6$  para  $x_i \in \mathbf{x}_d$  e solução robusta  $x_i \in [0.1, 0.3]$  para  $x_i \in \mathbf{x}_p$ . Uma forma de se verificar a robustez ou não de uma solução é observar o comportamento de cada conjunto de solução com a presença de ruído. Para isto, um ruído com distribuição uniforme no intervalo  $[-0.1, 0.1]$  foi adicionado a cada vetor solução  $\mathbf{x}_d$  ótima e robusta do problema pd-MOP02, gerando desta forma quatro soluções:

- um conjunto de soluções ótimas não robustas sem ruído da forma  $S_1 = \{\mathbf{x} \in \mathbb{R}^N; \mathbf{x} = (\mathbf{x}_p, \mathbf{x}_d)\}$  com  $x_i \in [0, 1]$  se  $x_i \in \mathbf{x}_p$  e  $x_j = 0.6$  se  $x_i \in \mathbf{x}_p$
- um conjunto de soluções ótimas não robustas com ruído da forma  $S_2 = \{\mathbf{x} \in \mathbb{R}^N; \mathbf{x} = (\mathbf{x}_p, \mathbf{x}_d)\}$  com  $x_i \in [0, 1]$  se  $x_i \in \mathbf{x}_p$  e  $x_j = 0.6 + \sigma$  se  $x_i \in \mathbf{x}_p$ , com  $\sigma_j \in \mathcal{U}(-0.1, 0.1)$ .

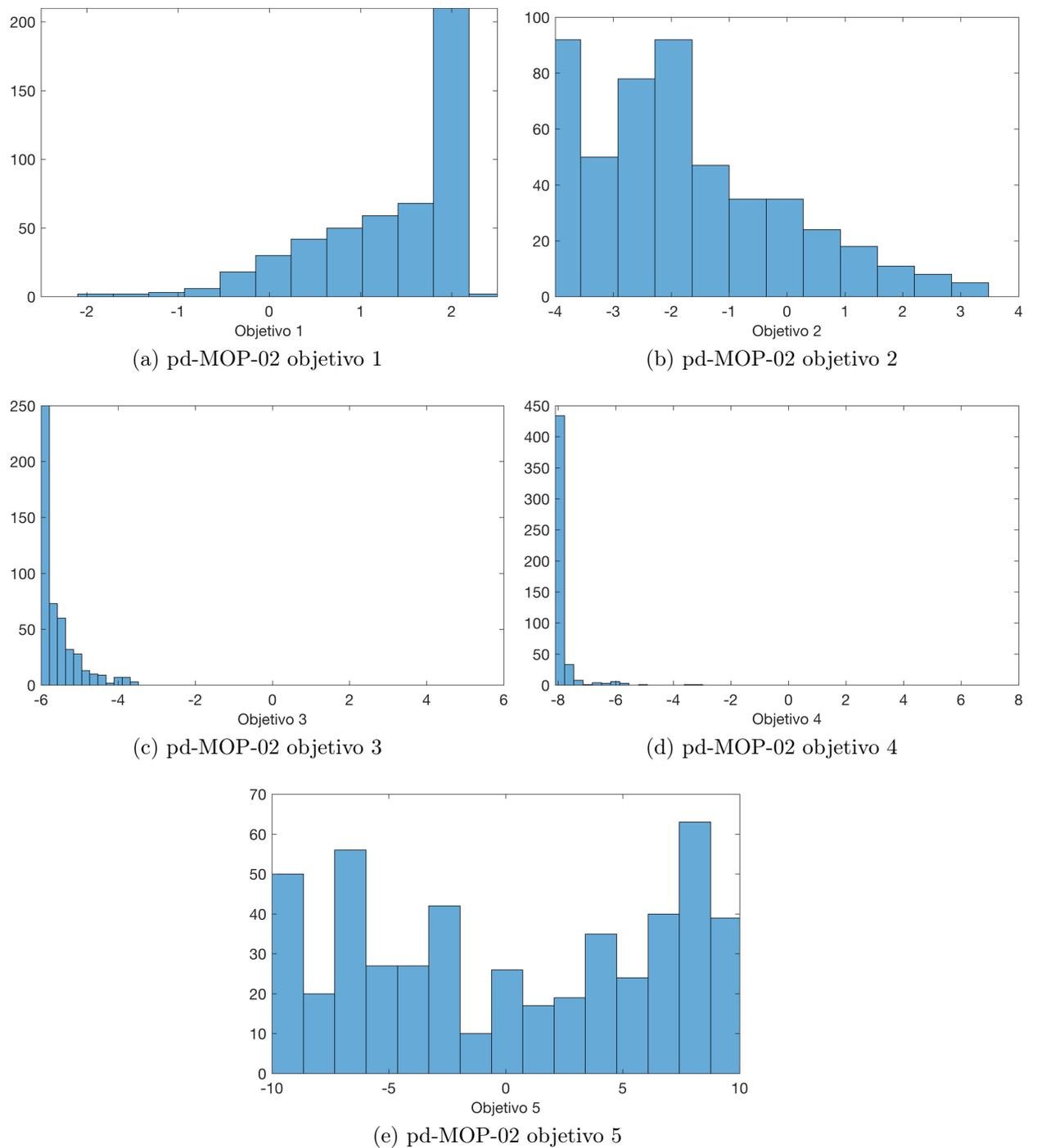


Figura 35 – Histograma das soluções obtidas no problema dissimilar pd-MOP-02 com 5 objetivos

- um conjunto de soluções ótimas robustas sem ruído da forma  $S_3 = \{\mathbf{x} \in \mathbb{R}^N; \mathbf{x} = (\mathbf{x}_p, \mathbf{x}_d)\}$  com  $x_i \in [0, 1]$  se  $x_i \in \mathbf{x}_p$  e  $x_j = 0.2$  se  $x_j \in \mathbf{x}_d$
- um conjunto de soluções ótimas robustas com ruído da forma  $S_4 = \{\mathbf{x} \in \mathbb{R}^N; \mathbf{x} = (\mathbf{x}_p, \mathbf{x}_d)\}$  com  $x_i \in [0, 1]$  se  $x_i \in \mathbf{x}_p$  e  $x_j = 0.2 + \sigma$  se  $x_j \in \mathbf{x}_d$ , com  $\sigma_j \in \mathcal{U}(-0.1, 0.1)$ .

Em cada solução dos conjuntos  $S_1$  a  $S_4$  definidos acima, vetor  $\mathbf{x}_p$  que determina a posição de cada solução no espaço dos objetivos não foi alterado. Desta forma a presença de ruído nas variáveis do problema irá alterar apenas a norma de cada vetor no espaço dos objetivos.

A figura 36 apresenta os resultados obtidos nesta simulação. Na figura 36a os elementos azuis representam as soluções ótimas não robustas definidas no conjunto  $S_1$ . Na trilha C de todos os setores o mapeamento angular representa também, utilizando círculos vermelhos, as soluções ótimas não robustas com ruído definidas no conjunto  $S_2$ . Como em cada solução o componente definido pelo vetor  $\mathbf{x}_p$  é o mesmo as soluções estão alinhadas, formando um padrão radial. Observe que a presença do ruído deteriorou a convergência das soluções.

Já a figura 36b apresenta as soluções ótimas robustas definidas pelos conjuntos  $S_3$  e  $S_4$ , sendo os elementos azuis pertencentes ao conjunto das soluções robustas sem ruído (conjunto  $S_3$ ). As soluções ótimas robustas com ruído presentes no conjunto  $S_4$  são representadas na trilha C de todos os setores, utilizando o mapeamento angular. Como há uma variação mínima entre estas soluções, uma ampliação da trilha C do setor 1 é apresentada na figura 36c.

Observe que as soluções ótimas robustas tem norma maior do que as soluções ótimas, o que pode ser verificado pela escala “vertical” do mapeamento angular (trilha C). Isto se deve ao fato de as soluções ótimas robustas serem soluções sub-ótimas globais. Entretanto, as soluções ótimas com ruído, definidas pelo conjunto  $S_2$ , têm uma norma maior do que as soluções ótimas robustas com ruído (definidas pelo conjunto  $S_4$ ) uma vez que a presença do ruído nas variáveis aumentou desproporcionalmente o valor nominal de cada objetivo.

### 5.2.3 Considerações finais

A visualização de dados em espaços de elevada dimensionalidade é um grande desafio para pesquisadores de todas as áreas. Os principais métodos utilizados para visualização de soluções de problemas multiobjetivo e com muitos objetivos apresentam virtudes e deficiências relacionadas à perda de informação causada pela utilização de suportes bidimensionais para a representação de informação com mais de duas dimensões. O método proposto não está imune a esta perda de informação mas procura compensar

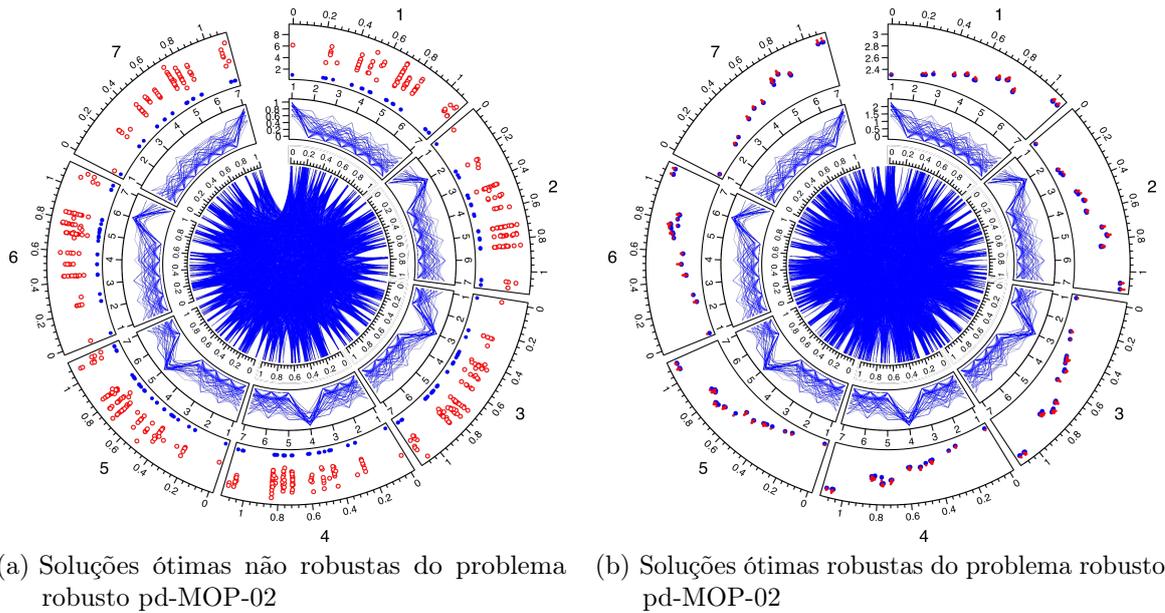


Figura 36 – CAP-vis aplicado ao problema robusto pd-MOP-02 com 7 objetivos

esta deficiência com a utilização combinada de diferentes métodos que juntos procuram contornar a limitação apresentada por cada um individualmente. Por exemplo, uma qualidade presente no mapeamento angular é a preservação da posição relativa de pontos alinhados com um vetor, e sua distância angular em relação aos eixos da base canônica do espaço  $\mathbb{R}^M$ . Esta característica não está presente em nenhum outro método analisado. Entretanto nesta técnica mais de um ponto do espaço de origem fornece o mesmo conjunto de coordenadas locais  $[\rho, \theta, j]$  e portanto estes pontos serão representados por um único ponto corresponde a um mesmo ponto no gráfico de coordenadas angulares. Esta deficiência é contornada nos gráficos de coordenadas paralelas e no diagrama de cordas. Entretanto, apesar destes representarem de forma clara a distribuição dos valores das coordenadas dos pontos em cada eixo, eles falham em fornecer a posição relativa do ponto no espaço.

Nas situações analisadas o método proposto foi capaz de fornecer informações sobre a distribuição espacial dos pontos, revelando detalhes que até então eram omitidos pelos

outros método ou acessíveis apenas por meio de alguma métrica especializada, o que faz desta ferramenta um importante aliado do tomador de decisão.

## 6 Geração de vetores de peso em algoritmos evolutivos

### 6.1 Introdução

Um dos motivos pelo qual os algoritmos de otimização multiobjetivo que utilizam o método da decomposição/agregação são populares é sua capacidade de estabelecer *a priori* um posicionamento “ideal” das soluções no primeiro ortante do espaço dos objetivos  $\mathbb{R}^M$  [63]. Este posicionamento é estabelecido pelos vetores de peso (*weight vectors*), também chamados de vetores de direção, que são um conjunto  $W = \{\mathbf{w}_1, \dots, \mathbf{w}_t\}$  de  $t$  vetores definido no início do algoritmo. Este conjunto de vetores faz a decomposição do primeiro ortante do espaço dos objetivos<sup>1</sup> em um conjunto de direções de modo que o algoritmo encontre pelo menos uma solução no espaço dos objetivos na região definida pelos vetores de peso. Neste sentido o nome “vetores de direção” é mais apropriado. Como muitas vezes esta é uma questão puramente semântica e as duas denominações são adequadas, este trabalho irá se referir ao conjunto  $W$  como conjunto de vetores de peso. Uma exceção à ambiguidade entre vetores de peso e vetores de direção é apresentada por Jiao et al. [66], que apresenta um algoritmo (DVCMOEA) que é derivado do MOEA/D que utiliza vetores de direção  $\lambda_i$  com  $|\lambda_i|_2 = 1$  e faz a distinção entre este tipo de vetor e um vetor de peso  $\mathbf{w}_i$ , com  $|\mathbf{w}_i|_1 = 1$ .

O algoritmo evolutivo faz o posicionamento das soluções nas direções definidas pelo conjunto de vetores  $W$  utilizando as funções de agregação  $f(\mathbf{x}, \mathbf{w}_i)$ . Estas funções fazem a decomposição do problema de otimização multiobjetivo original em  $k$  problemas de otimização mono-objetivo. Cada uma das funções  $f(\mathbf{x}, \mathbf{w}_i)$  procura um ponto  $F(\mathbf{x})$  no espaço dos objetivos que esteja na região definida pelo vetor  $\mathbf{w}$ . Cada função de agregação procura uma solução de compromisso entre a norma do ponto  $F(\mathbf{x})$  e seu alinhamento com o vetor  $\mathbf{w}_i$ . Neste sentido o nome “vetores de peso” é mais apropriado uma vez que  $\mathbf{w}_i$  é um parâmetro (ou peso) da função de agregação.

O conjunto de vetores  $W$  é característico da dimensão  $M$  do espaço dos objetivos. Diferentes problemas com  $M$  objetivos podem utilizar o mesmo conjunto de vetores de peso  $W$ , que de maneira geral se mantêm inalterados no decorrer do processo de otimização. Entretanto alguns autores [83, 25, 119, 159] propõem a atualização dinâmica destes vetores durante o processo de otimização.

De maneira geral os vetores do conjunto  $W$  constituem uma amostra de pontos distribuídos na porção contida no primeiro ortante do espaço  $\mathbb{R}^M$  do hiper-plano  $x_1 + \dots +$

<sup>1</sup> No caso dos problemas dissimilares os objetivos são normalizados no intervalo  $[0, 1]$ .

$x_M = 1$  e que apresentem uma baixa medida de discrepância dos pontos [165, 2, 112]. O método mais popular de geração destes pontos foi proposto por Scheffé [130] que define um  $\{M, q\}$ -simplex utilizado em experimentos com misturas. Neste método  $\binom{M+q-1}{q}$  pontos  $\mathbf{x} \in \mathbb{R}^M$  são gerados com  $q + 1$  pontos igualmente espaçados no bordo do simplex e satisfazendo  $|\mathbf{x}|_1 = x_1 + \dots + x_M = 1$  que gera um conjunto de vetores igualmente distribuídos. Posteriormente Das and Dennis [24] apresentam um método construtivo para a obtenção do conjunto  $W$  utilizando o método de Scheffé. Liu and Li [88] apresentam um método construtivo para a obtenção de vetores de peso igualmente espaçados utilizando coordenadas esféricas. Entretanto alguns autores [44, 145, 92, 31] utilizam um conjunto de vetores de peso com distribuição não homogênea.

O método de Scheffé cria rapidamente o conjunto de vetores  $W$  mas apresenta uma série de limitações:

1. O conjunto de vetores  $W$  apresenta muitos elementos, especialmente em problemas com mais do que três objetivos. Com isto, o tamanho da população dos algoritmos também fica muito grande, pois o número de elementos de  $W$  define o tamanho da população utilizado no algoritmo.
2. Este método gera muitos vetores de peso  $\mathbf{w}_i$  no bordo do simplex, que é uma região de pouco interesse principalmente em problemas com mais do que três objetivos. De fato, se uma solução está no bordo do simplex pelo menos um dos objetivos tem um valor  $\epsilon$  próximo de zero. Caso esta situação seja de interesse do tomador de decisão é mais eficiente colocar este objetivo como uma restrição do problema, pois desta forma se reduz a dimensão do espaço dos objetivos tornando o processo de busca mais eficiente [79].
3. Os vetores de  $W$  não têm a mesma norma euclidiana. Desta forma, os métodos de agregação que utilizam alguma norma ponderada (como o método da soma ponderada ou da norma de Tchebycheff) apresentam resultados diferentes para pontos do espaço dos objetivos com características semelhantes.

## 6.2 Método proposto

Para contornar as limitações apresentadas, acima este trabalho apresenta um novo método de geração de vetores de peso [98], utilizando uma estratégia evolutiva  $ES(\mu + 1, \mu)$ . O método proposto permite a criação de um conjunto  $W$  no primeiro ortante do espaço  $\mathbb{R}^M$ , localizados em uma hiper-superfície de dimensão  $M - 1$  de  $p$ -norma constante. O número de vetores do conjunto  $W$  e o valor de  $p$  da  $p$ -norma dos vetores podem ser definidos arbitrariamente pelo tomador de decisão, que também define a distância entre o conjunto de vetores e o bordo do primeiro ortante do espaço  $\mathbb{R}^M$ . Caso seja de interesse

do tomador de decisão, esta distância pode ser definida como sendo igual a zero e desta forma são gerados vetores contidos no bordo do ortante. Os vetores não estão igualmente espaçados, mas quase igualmente espaçados e desta forma  $W$  é um conjunto de vetores de baixa discrepância. Além da criação de um conjunto de vetores de peso, o método proposto permite a criação de um cone de vetores de peso, que consiste em um conjunto de vetores que define uma região específica do espaço dos objetivos. A definição do cone de vetores de peso é simples e intuitiva e sua implementação exige uma pequena modificação no algoritmo de geração de vetores de peso usuais.

O método proposto inicia com a geração de uma população inicial  $W = \{\mathbf{w}_1, \dots, \mathbf{w}_t\}$  de  $t$  vetores no hiper-cubo  $[0, 1]^M$  do espaço  $\mathbb{R}^M$  utilizando uma amostra distribuída em um hiper-cubo latino [141, 160] e que são normalizados de modo que sua  $p$ -norma seja igual a um, fazendo  $\mathbf{w} = \frac{\mathbf{w}}{\|\mathbf{w}\|_p}$ . Após esta normalização é feito um ajuste em suas coordenadas de modo a evitar a presença de vetores no bordo do primeiro ortante do espaço  $\mathbb{R}^M$ . Em seguida é calculada a matriz de distâncias  $d_{i,j}$  com a distância euclidiana entre os vetores distintos  $\mathbf{w}_i$  e  $\mathbf{w}_j$ , com  $1 \leq i, j \leq t$  e  $i \neq j$ . Aleatoriamente é escolhido um  $\mathbf{w}_i \in W$  e é gerado um novo vetor  $\mathbf{w}'_i$  pela adição de um vetor de perturbação gaussiana  $\delta_i$  de média zero e variância  $\sigma^2$ , ou seja,  $\mathbf{w}'_i = \mathbf{w}_i + \delta_i$  com  $\delta_i = (\delta_{i,1}, \dots, \delta_{i,N})$  e  $\delta_{i,j} \in \mathcal{N}(0, \sigma^2)$  e é calculada a distância euclidiana  $d_j$  entre o novo vetor  $\mathbf{w}'_i$  e os demais vetores  $\mathbf{w}_j \in W$ ,  $1 \leq j \leq t$ . A atualização da população é feita da seguinte maneira: é removido o vetor do conjunto  $W$  que maximiza a menor distância entre o novo vetor  $\mathbf{w}'_i$  e os demais vetores  $\mathbf{w}_j \in W$ , seguindo uma estratégia  $ES(\mu + 1, \mu)$ . A função de aptidão  $\phi$  que guia este processo evolutivo é a soma das distâncias entre os  $k$  vizinhos mais próximos.

O pseudo-código 4 apresenta o fluxo do algoritmo de geração de vetores de peso e em seguida são apresentados os detalhes deste método de geração de vetores de peso.

**Inicialização:** Nesta etapa os parâmetros iniciais são definidos.

1. Defina o número  $t$  de vetores de peso a serem criados
2. Defina o valor de  $p$  da  $p$ -norma que será utilizada.
3. Defina o valor do *gap*  $g$ . A fim de evitar a presença de vetores de peso próximos do bordo do primeiro ortante, as coordenadas  $w_i$  de qualquer vetor  $\mathbf{w} = (w_1, \dots, w_t)$  na população inicial  $W = \{\mathbf{w}_1, \dots, \mathbf{w}_n\}$  e dos demais vetores criados devem satisfazer

$$g \leq w_i \leq 1 - g \quad (6.1)$$

Se  $w_i < g$  faça  $w_i = g$  e se  $w_i > 1 - g$  faça  $w_i = 1 - g$ . Fazendo  $g = 0$  o método produzirá vetores de peso no bordo do primeiro ortante do espaço  $\mathbb{R}^M$  da mesma maneira que o método de Scheffé.

4. Defina o número máximo de iterações *itermax* e o número de vizinhos  $K$  utilizado na função de aptidão. Após a realização de alguns testes o valor de  $k = 2$  é sugerido.

**Algoritmo 4:** Geração de vetores de peso

---

**input** : O número de vetores de peso  $t$ , o valor de  $p$  da  $p$ -norma, o valor do  $gap$   $g$ , o número de vizinhos  $K$  e o número máximo de iterações.

**output** : O conjunto  $W$  de  $t$  vetores de peso.

```

/* Inicialização */
1 iter ← 0;
2 W = {w1, ..., wt} ; /* Inicialize a população de tamanho t */
3 wi ←  $\frac{\mathbf{w}_i}{|\mathbf{w}_i|_p}$  ; /* Normalize a população W de modo que |w|p = 1 */
4 wi ← gap(wi) ; /* Ajuste as coordenadas wi dos vetores de peso de acordo com o valor do gap g */
5 di,j ← D(wi, wj) ; /* Calcule a distância entre wi e wj */
6 φ(wi) ; /* Avalie a função de aptidão para wi ∈ W */
/* Ciclo evolutivo */
7 while iter < itermax do
8   Escolha i em {1, ..., n} aleatoriamente;
9   w'i ← wi + δ ;
10  d'j ← D(w'i, wj) ; /* Avalie a distância entre w'i e wj ∈ W */
11  φ(w'i) ; /* Avalie a função de aptidão de w'i */
12  Substitua o vetor pior avaliado wj de W por w'i;
13  iter ← iter + 1;
14 end while

```

---

Após estabelecer os parâmetros iniciais, gere a população inicial  $W$  por meio de uma amostra em um cubo latino. Normalize os vetores de  $W$  e verifique se suas coordenadas satisfazem a inequação (6.1), fazendo algum ajuste caso seja necessário. Depois calcule a matriz de distância  $d_{i,j}$  entre os pares de vetores  $\mathbf{w}_i$  e  $\mathbf{w}_j$  e a função de aptidão de cada vetor  $\mathbf{w}_i$ . A função de aptidão do vetor  $\mathbf{w}_i \in W$  é dada pela soma das distâncias de  $\mathbf{w}_i$  aos seus  $K$  vizinhos mais próximos em  $W$ .

**Ciclo Evolucionário:** Enquanto o critério de parada não for atingido repita os passos a seguir:

1. Escolha aleatoriamente um vetor  $\mathbf{w}_i \in W$ .
2. Gere o novo vetor

$$\mathbf{w}'_i = \mathbf{w}_i + \delta_i \quad (6.2)$$

O vetor de perturbação  $\delta_i$  é definido da seguinte maneira:

- a) Determine a menor distância  $d_{i,\min}$  entre  $\mathbf{w}_i$  e os outros vetores  $\mathbf{w}_j \in W$ ,  $i \neq j$ .
- b) Calcule a penalidade  $\alpha = (1 - t)a + tb$  com  $t = \frac{iter}{itermax}$ . Em nossos testes foi utilizado  $a = 1.5$  e  $b = 0.1$ .
- c) Calcule  $\delta_i = (\delta_{i1}, \dots, \delta_{im})$ , onde  $\delta_{ij}$  é uma variável aleatória com distribuição normal de média zero e variância  $\sigma_i^2 = \alpha \times d_{i,\min}$ , ou seja,  $\delta_{ij} \sim N(0, \sigma_i^2)$ . Esta

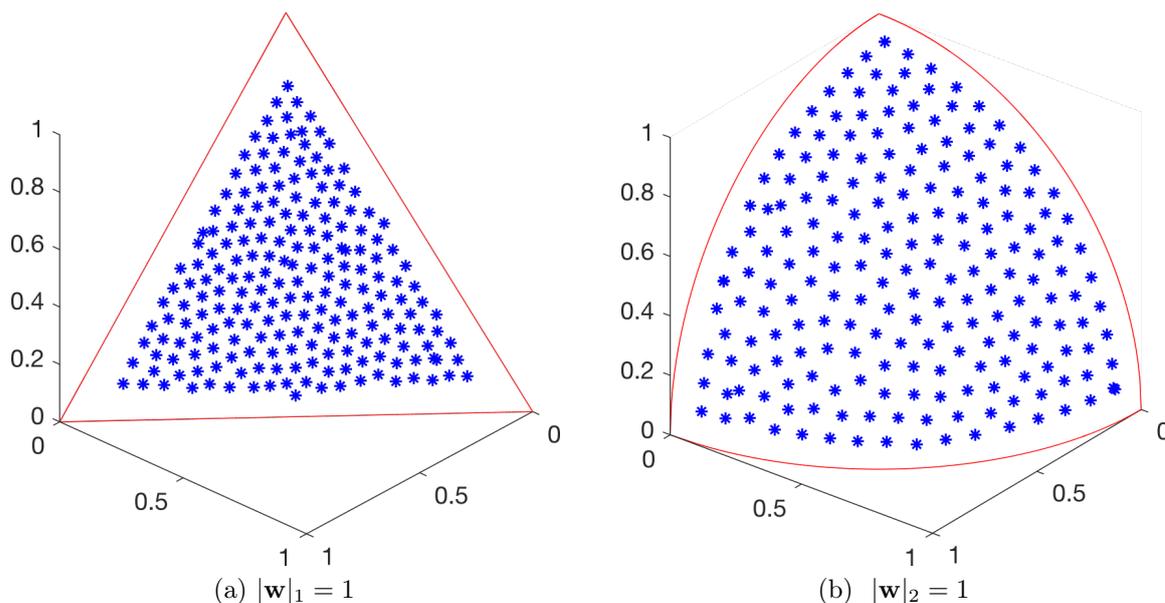


Figura 37 – Dois conjuntos de vetores de peso no espaço  $\mathbb{R}^3$  gerados pelo método proposto. A linha vermelha representa a interseção das superfícies de  $p$ -norma constante com o bordo do primeiro ortante.

escolha permite a adaptação da mutação de acordo com a vizinhança do ponto. Além disso ela favorece a exploração do espaço de busca no início do algoritmo e a busca local no final do processo.

d) Normalize o novo vetor  $\mathbf{w}_i = (w'_{i,1}, \dots, w'_{i,t})$  e verifique as condições  $g \leq w'_{ij} \leq 1 - g$ . Se  $w'_{ij} < g$  faça  $w'_{ij} = g$  e se  $w'_{ij} > 1 - g$  faça  $w'_{ij} = 1 - g$ .

3. Calcule a distância euclidiana entre  $\mathbf{w}'_i$  e os demais vetores  $\mathbf{w}_j \in W$ .

4. Calcule a função de aptidão  $\phi(\mathbf{w}_j)$ ,  $\mathbf{w}_j \in W$ .

5. Seja

$$\mathbf{w}_{min} = \arg \min_j \phi(\mathbf{w}_j), \quad \mathbf{w}_j \in W \quad (6.3)$$

Se  $\phi(\mathbf{w}'_i) > \phi(\mathbf{w}_{min})$  substitua  $\mathbf{w}_{min}$  por  $\mathbf{w}'_i$  em  $W$  e atualize a matriz de distâncias  $d_{i,j}$ . Caso contrário descarte  $\mathbf{w}'_i$ .

6. Atualize o contador de iterações.

A figura 37 apresenta dois conjuntos com  $t = 200$  vetores de peso cada um, utilizando  $gap$   $g = 0.1$  e valores de  $p$ -norma iguais a um (figura 37a) e dois (figura 37b). A linha vermelha apresenta a interseção das superfícies de  $p$ -norma constante com o bordo do primeiro ortante. Os vetores apresentados nestas figuras apresentam uma boa distribuição, sem a presença de agrupamentos de vetores ou espaços vazios. A distância do conjunto de vetores ao bordo do primeiro ortante é regular e bem definida.

### 6.2.1 Cone de vetores

Um cone de vetores de peso é um conjunto de vetores contidos na região interna de um cone. O cone é definido por meio de um vetor de preferências  $\mathbf{v}$  que corresponde ao eixo do cone e um ângulo  $\theta$  entre o eixo  $\mathbf{v}$  e sua geratriz. Estes elementos são ilustrados na figura 38c.

A geração de um cone de vetores no espaço  $\mathbb{R}^M$  requer poucas modificações no algoritmo de geração de vetores de peso: na fase de inicialização, especifique o eixo  $\mathbf{v}$  e o ângulo  $\theta$  que definem o cone de vetores. Após a etapa de normalização dos elementos da população calcule o ângulo  $\alpha_i$  entre um vetor  $\mathbf{w}_i$  e o eixo  $\mathbf{v}$  do cone. Se  $\alpha_i > \theta$  então o vetor  $\mathbf{w}_i$  está na região exterior do cone e desta maneira sua função de avaliação deve ser penalizada da seguinte maneira:

$$\phi_p(\mathbf{w}_i) = \phi(\mathbf{w}_i) - M \times \max\{\alpha_i - \theta, 0\} \quad (6.4)$$

sendo  $M$  a dimensão do espaço,  $\phi_p(\mathbf{w}_i)$  a função de aptidão penalizada e  $\phi(\mathbf{w}_i)$  a função de aptidão original.

A figura 38c ilustra os elementos de um cone, destacando seu eixo, definido pelo vetor  $\mathbf{v}$ , e o ângulo de abertura  $\theta$ . A figura 38a mostra um cone de 20 vetores no espaço  $\mathbb{R}^2$  definido pelo eixo  $\mathbf{v} = (0.3, 0.5)$  e ângulo  $\theta = \pi/10$  radianos, com norma euclidiana. Na figura 38b está presente um cone composto por 100 vetores no espaço  $\mathbb{R}^3$  utilizando a norma euclidiana, definido pelo eixo  $\mathbf{v} = (0.3, 0.3, 0.5)$  e ângulo  $\theta = \pi/20$  radianos,

## 6.3 Análise do método proposto

A fim de verificar a aplicabilidade do uso do conjunto de vetores de peso gerados pela nova metodologia dois experimentos foram realizados. O primeiro experimento procura quantificar a qualidade do conjunto de vetores de peso gerados pelo método proposto enquanto que o segundo experimento compara soluções obtidas pelos algoritmos MOEA/D e NSGAIII utilizando vetores de peso gerados pelo método de Scheffé e pelo método proposto.

### 6.3.1 Experimento 1

Para verificar a regularidade da dispersão dos vetores de peso de um conjunto  $W$  foi calculada a distância entre cada vetor  $\mathbf{w}_i$  e seu vizinho mais próximo em três conjuntos de vetores de peso com mesma característica, gerados de forma independente. Em seguida foi calculada a média  $\mu$ , o desvio padrão  $\sigma$  e o coeficiente de variação de Pearson  $CV = \sigma/\mu$  deste conjunto de distâncias. Também conhecido como desvio padrão relativo, o coeficiente de variação de Pearson mostra a extensão da variabilidade em relação

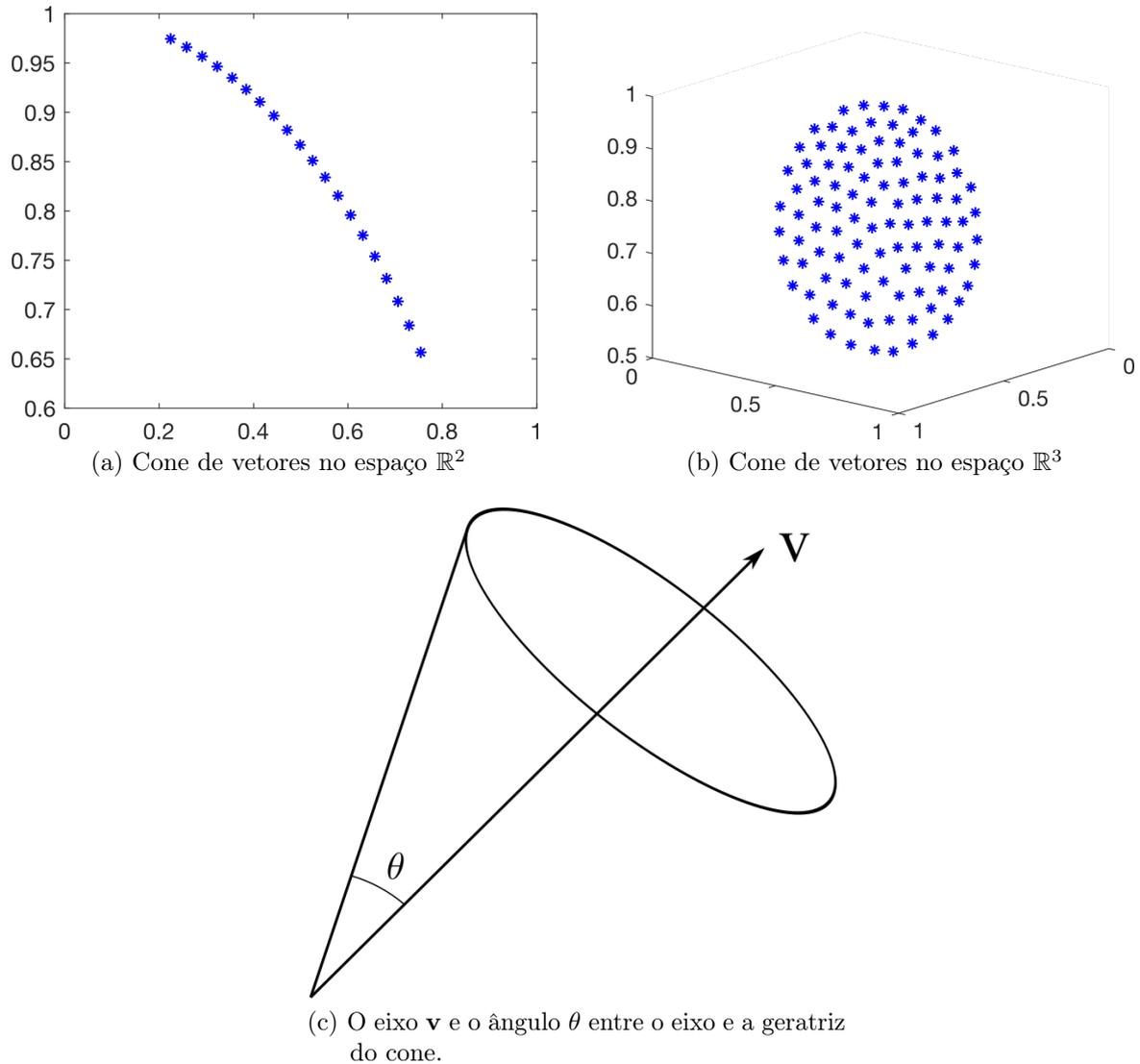


Figura 38 – Os elementos de um cone e dois conjuntos de cones de vetores nos espaços  $\mathbb{R}^2$  e  $\mathbb{R}^3$

Dimensão ( $M$ ):	5, 10, 20
Número de vetores ( $t$ ):	250, 500, 750, 1000
$gap$ :	0.05
$p$ -norma:	2
número de iterações:	6000

Tabela 1 – Parâmetros dos vetores de peso gerados nos experimentos realizados

à média da população. Em um conjunto de vetores igualmente distribuídos se tem que  $\sigma = 0$  e portanto  $CV = 0$ . Em uma distribuição aleatória de pontos o coeficiente de variação é maior do que 0.5.

As características dos conjuntos de vetores de peso utilizados neste experimento são apresentadas na tabela 1 e os valores  $\mu$ ,  $\sigma$  e  $CV$  obtidos são apresentados na tabela 2. Estes resultados indicam uma distribuição homogênea dos vetores de peso gerados pelo

Dimensão	Pontos	Média $\mu$	Desvio padrão $\sigma$	CV
$\mathbb{R}^5$	250	0.2493	0.0118	0.0472
		0.2501	0.0140	0.0559
		0.2514	0.0114	0.0452
	500	0.1945	0.0124	0.0635
		0.1961	0.0099	0.0503
		0.1962	0.0107	0.0547
	750	0.1698	0.0106	0.0623
		0.1700	0.0102	0.0602
		0.1696	0.0105	0.0620
	1000	0.1532	0.0102	0.0663
		0.1537	0.0103	0.0672
		0.1540	0.0101	0.0655
$\mathbb{R}^{10}$	250	0.5210	0.0145	0.0278
		0.5233	0.0145	0.0277
		0.5259	0.0150	0.0286
	500	0.4515	0.0138	0.0305
		0.4531	0.0142	0.0312
		0.4524	0.0144	0.0318
	750	0.4141	0.0145	0.0351
		0.4148	0.0138	0.0334
		0.4149	0.0155	0.0373
	1000	0.3911	0.0146	0.0375
		0.3900	0.0141	0.0363
		0.3909	0.0144	0.0367
$\mathbb{R}^{20}$	250	0.7374	0.0117	0.0159
		0.7422	0.0135	0.0182
		0.7389	0.0146	0.0197
	500	0.6792	0.0135	0.0198
		0.6780	0.0137	0.0202
		0.6777	0.0134	0.0197
	750	0.6437	0.0142	0.0221
		0.6433	0.0143	0.0222
		0.6433	0.0148	0.0230
	1000	0.6202	0.0145	0.0234
		0.6212	0.0145	0.0234
		0.6201	0.0151	0.0243

Tabela 2 – Média, desvio padrão e coeficiente de variação da distância entre um vetor de peso  $\mathbf{w}_i$  e seu vizinho mais próximo em espaços de dimensão 5, 10 e 20.

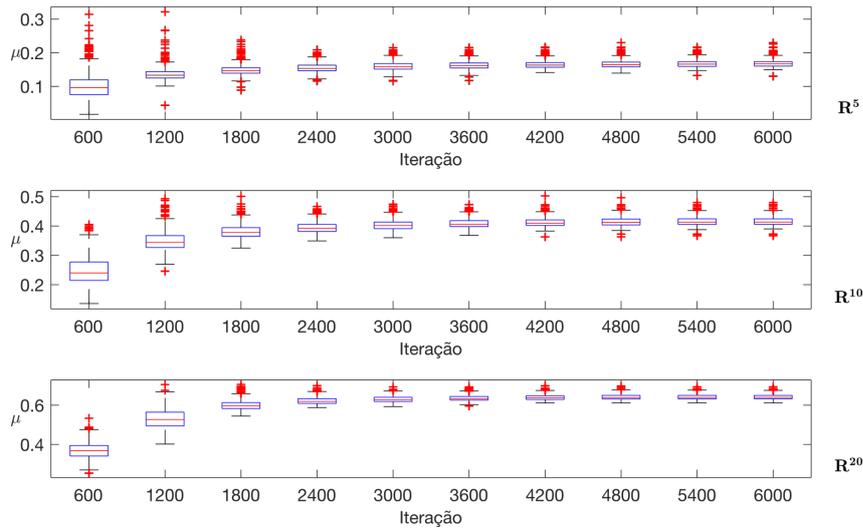


Figura 39 – Boxplot da evolução do valor médio da distância entre um vetor de peso  $\mathbf{w}_i$  e seu vizinho mais próximo em espaços de dimensão 5,10 e 20.

método proposto. Em todos os casos o valor médio da distância entre um vetor e seu vizinho mais próximo diminui com o aumento do número de vetores gerados e o desvio padrão  $\sigma$  é pequeno, com  $0.01 < \sigma < 0.015$  exceto no segundo conjunto de vetores no espaço  $\mathbb{R}^5$  com 500 vetores onde  $\sigma = 0.0099$  e o terceiro conjunto de vetores no espaço  $\mathbb{R}^{20}$  com 1000 vetores onde  $\sigma = 0.0151$ .

A evolução do valor médio da distância entre um vetor ao seu vizinho mais próximo nos espaços de dimensão 5,10 e 15 é ilustrada na figura 39. Os valores representados nesta figura seguem um padrão bem definido: Nas primeiras iterações a distância média é menor e seu valor se estabiliza por volta da iteração 2.000. Além disto o intervalo interquartil também se reduz com o aumento do número de iterações. Em todos os blocos estão presentes alguns *outliers*, principalmente acima do limite superior, mas sua quantidade decresce, bem como sua distância até a mediana.

Uma outra forma de verificar a qualidade da distribuição dos vetores de peso obtidos pelo método proposto é por meio da sua comparação direta com um conjunto de vetores gerados pelo método de Scheffé. Como neste método os vetores apresentam uma distribuição regular não existe a formação de agrupamentos de vetores ou regiões do espaço menos povoadas do que outras. Para fazer esta comparação três conjuntos de vetores de peso foram gerados pelo método de Scheffé em espaços de dimensão 5, 10 e 20. O número de vetores e os parâmetros utilizados em cada conjunto de vetores é apresentado na tabela 3.

Para cada conjunto de vetores apresentado na tabela 3 foi gerado um conjunto com a mesma quantidade de vetores de peso pelo método proposto e estes conjuntos foram comparados pelo método agrupamento hierárquico (*hierarchical clustering method* - [53]). Neste método dois agrupamentos de pontos próximos em uma dada iteração são

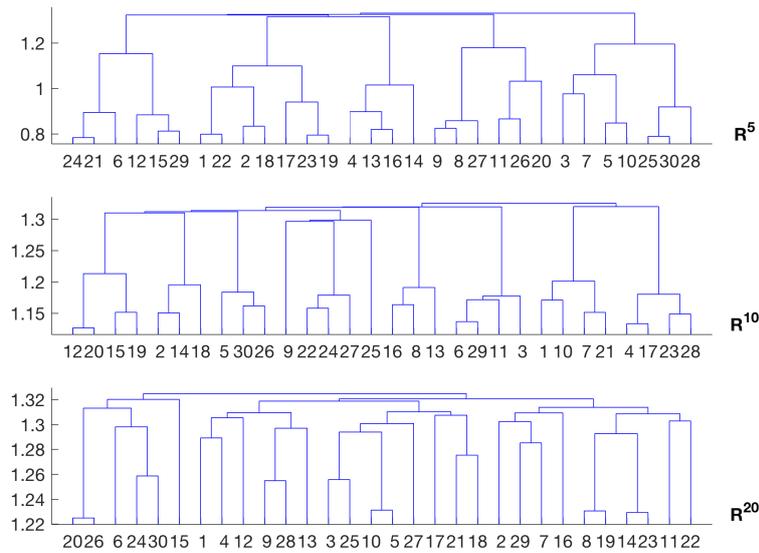


Figura 40 – Dendrograma do processo de formação de agrupamentos de vetores nos espaços  $\mathbb{R}^5$ ,  $\mathbb{R}^{10}$  e  $\mathbb{R}^{20}$ .

reunidos em um agrupamento maior até que todos os elementos estejam agrupados em um único grupo. Foi utilizado o método *complete linkage* como medida de distância entre agrupamentos. Em cada etapa do processo de aglomeração dos pontos a distância de fusão utilizada para se definir a união de dois agrupamentos foi anotada. A distância de fusão informa a distância mínima entre dois agrupamentos em cada iteração. Com estes valores foi construído os dendrogramas apresentados na figura 40, que representam o processo de formação de agrupamentos.

A figura 41 apresenta os valores da distância de fusão encontrada ao se fazer os agrupamentos de vetores de peso gerados pelo método proposto nos espaços  $\mathbb{R}^5$ ,  $\mathbb{R}^{10}$  e  $\mathbb{R}^{20}$ . Uma vez que os vetores gerados pelo método de Scheffé apresentam uma distribuição regular, sua distância de fusão segue um padrão de escada, com valores bem definidos. Nos conjuntos de vetores gerados pelo método proposto esta distância de fusão segue uma linha suave, porém próxima da curva em formato de escada típica dos vetores gerados pelo método de Scheffé, o que indica uma similaridade entre as distribuições.

A tabela 4 faz a comparação da distância entre um vetor de peso  $\mathbf{w}_i$  e seu vizinho

dimensão	pontos no bordo	número de vetores
5	10	1001
10	5	2002
20	3	1540

Tabela 3 – Parâmetros utilizados na geração de vetores pelo método de Scheffé

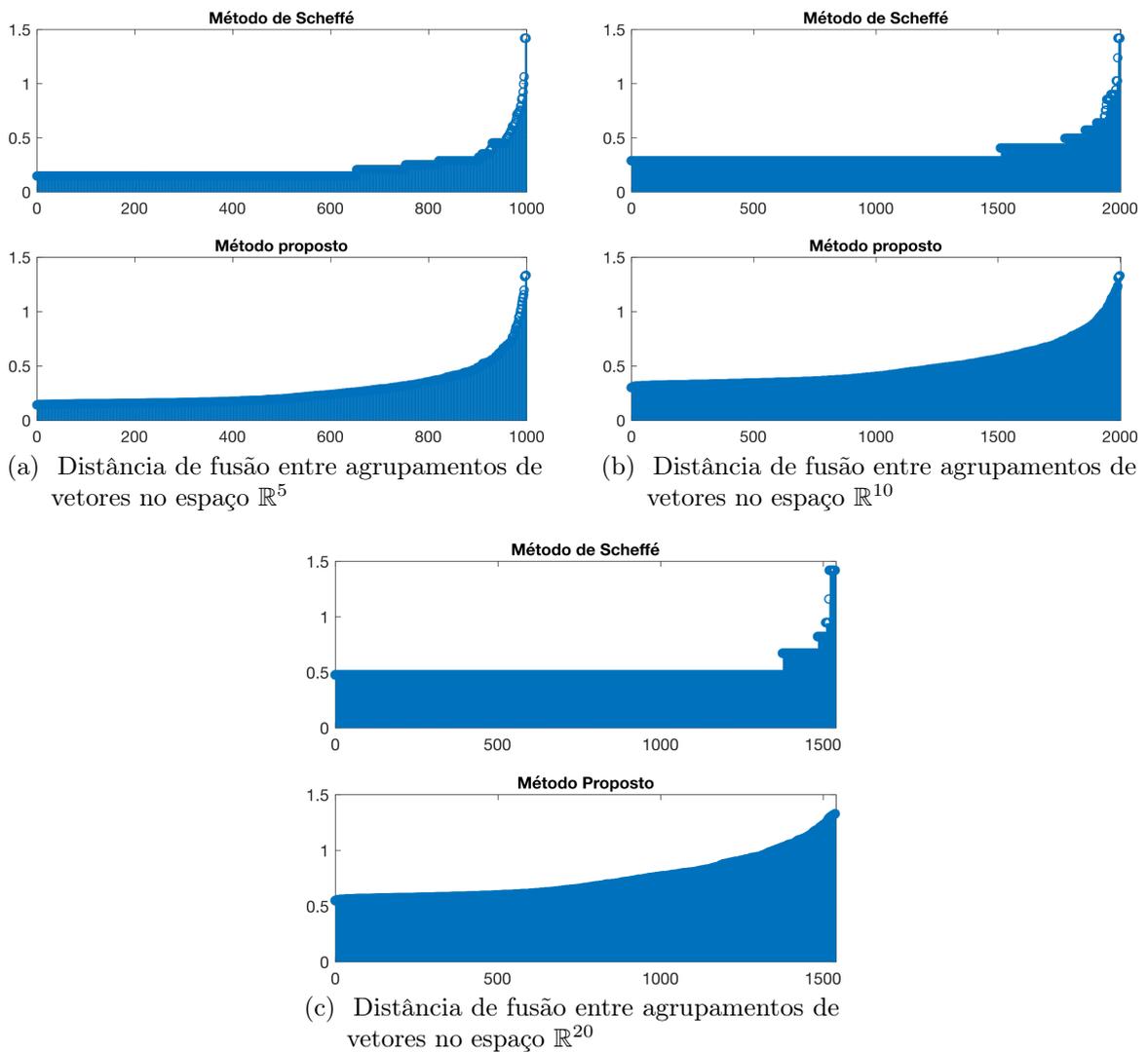


Figura 41 – Distância de fusão entre agrupamentos de vetores nos espaços  $\mathbb{R}^5$ ,  $\mathbb{R}^{10}$  e  $\mathbb{R}^{20}$ . A escala horizontal indica o ordinal do conjunto de vetores de peso.

mais próximo, apresentando o valor médio, o desvio padrão e o coeficiente de variação de Pearson desta medida utilizando um conjunto de vetores gerados pelo método de Scheffé e o conjunto de vetores analisado pelo método de agrupamento hierárquico. Observe que no método proposto a distância média entre vetores é menor, com desvio padrão e coeficiente de variação baixos.

A figura 42 faz uma comparação entre um conjunto de 2002 vetores de peso no espaço  $\mathbb{R}^{10}$  gerados pelo método de Scheffé (figura 42a) e o método proposto (figura 42b) utilizando o CAP-vis. Os vetores gerados pelo método de Scheffé formam um padrão regular, o que pode ser observado pela simetria do gráfico de diagrama de cordas na trilha A, pela regularidade do gráfico de coordenadas paralelas na trilha B e pela sobreposição dos pontos representados pelo mapeamento angular na trilha C do gráfico 42a. Como os vetores gerados pelo método proposto não apresentam este padrão geométrico, a representação destes pontos não apresenta qualquer tipo simetria ou regularidade. Entretanto, como pode

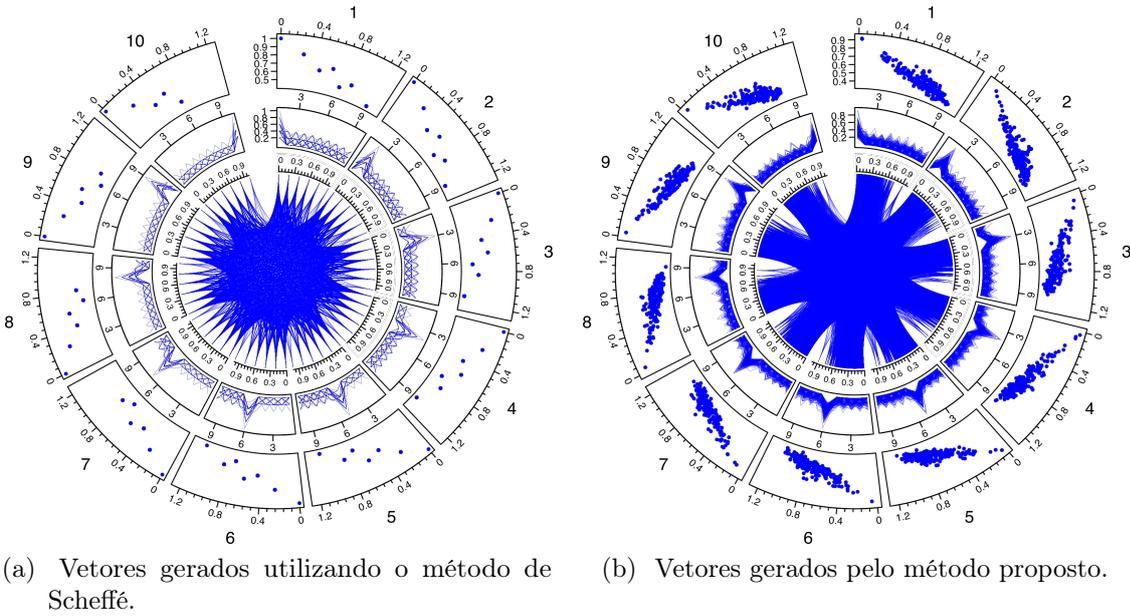


Figura 42 – Dois conjuntos de vetores de peso no espaço  $\mathbb{R}^{10}$  gerados por diferentes métodos.

ser observado no mapeamento angular presente na trilha C, este método é capaz de gerar uma variedade mais rica de vetores, ocupando posições do espaço que eram negligenciadas pelo método de Shceffé, que privilegia a regularidade da distribuição dos vetores. Nos dois conjuntos todos os vetores de peso  $\mathbf{w}$  têm norma constante  $|\mathbf{w}|_1 = 1$ .

Para ilustrar a geração de cones em espaços de elevada dimensionalidade, dois cones de vetores de peso foram gerados no espaço  $\mathbb{R}^{10}$ , contendo 1000 vetores de peso cada um, considerando dois conjuntos de preferências definidos por um tomador de decisão. Estes dois conjuntos de cones de vetores são apresentados na figura 43 utilizando a ferramenta CAP-vis.

O primeiro cone tem como eixo o vetor a hiper-diagonal  $\mathbf{v}_1 = (1, \dots, 1)$  e ângulo de abertura  $\theta = \frac{\arccos(1/\sqrt{10})}{10}$ . Este conjunto de vetores reflete a preferência por um equilíbrio

dimensão	Método	média $\mu$	desvio padrão $\sigma$	CV
5	Scheffé	0.1414	0	0
	Proposto	0.1355	0.0089	0.0657
10	Scheffé	0.2828	0	0
	Proposto	0.1676	0.0132	0.0789
20	Scheffé	0.4714	0	0
	Proposto	0.2322	0.0138	0.0594

Tabela 4 – Média, desvio padrão e coeficiente de variação de Pearson da distância entre um vetor de peso  $\mathbf{w}_i$  e seu vizinho mais próximo.

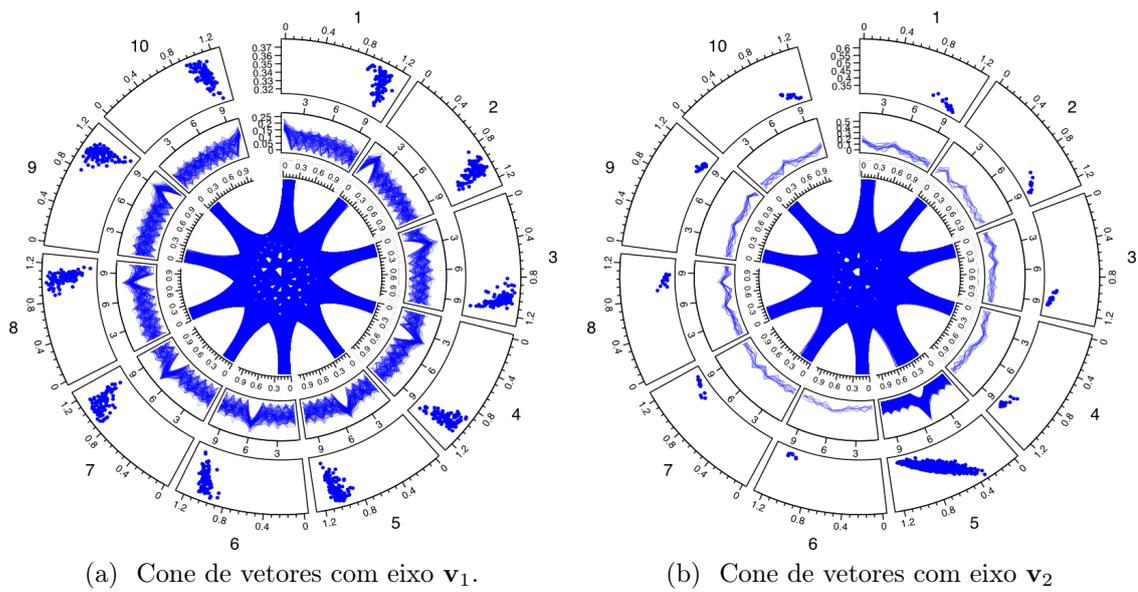


Figura 43 – Dois cones de vetores de peso no espaço  $\mathbb{R}^{10}$  com diferentes eixos.

entre os valores dos objetivos sem a presença de soluções extremas, onde alguns objetivos têm um valor muito alto e outros um valor muito baixo.

O segundo cone reflete um problema onde o objetivo 5 tem mais importância que os demais. Esta informação é incorporada na forma de pesos nas coordenadas do eixo  $\mathbf{v}_2$  do cone. Desta forma a quinta coordenada do vetor  $\mathbf{v}_2$  é igual a 3 e as demais são iguais a um, ou seja, o eixo do segundo cone de vetores é definido pelo vetor  $\mathbf{v}_2 = (1, 1, 1, 1, 3, 1, 1, 1, 1, 1)$ . Como esta segunda situação pretende explorar uma região mais específica do espaço dos objetivos, o ângulo de abertura deste cone é definido como  $\theta = \frac{\arccos(1/\sqrt{10})}{20}$ .

A figura 43a mostra o cone de vetores com eixo alinhado com a hiper-diagonal do espaço. O mapeamento angular presente na trilha C desta figura mostra que houve uma boa distribuição dos vetores de peso na região central do espaço dos objetivos com a presença de vetores em todos os setores do gráfico e não é observado vetores nas regiões periféricas, próximos ao bordo do primeiro ortante.

A figura 43b mostra o cone de vetores onde o tomador de decisão estabelece uma preferência pelo quinto objetivo. Nesta figura podemos observar que a maior parte dos vetores de peso está concentrada no setor 5 deste gráfico o que mostra que de fato o cone de vetores gerado reflete as características definidas pelo tomador de decisão. Observe que também estão presentes vetores nos demais setores, porém em menor quantidade. Estes pontos correspondem à porção do cone de vetores localizada na região central do espaço dos objetivos.

O uso dos cones de vetores em algoritmos de otimização que utilizam as técnicas de decomposição/agregação faz com o que as preferências do tomador de decisão sejam incorporadas no processo de otimização. Esta incorporação aliada ao método de visualização

CAP-vis facilita a análise do problema e a escolha da solução final, o que pode ser uma tarefa desafiadora se esta escolha se realizar em um conjunto irrestrito de soluções distribuídas em todo o espaço dos objetivos. e sem uma ferramenta de visualização adequada.

### 6.3.2 Experimento 2

Neste experimento foram utilizados os algoritmos MOEA/D e NSGAIII utilizando vetores de peso gerados pelo método de Scheffé e pelo método proposto em 30 instâncias independentes. Foi utilizado o problema de *benchmark* pdMOP02 não dissimilar e sem o uso da meta-variável  $\mathbf{y}$  descrita na equação (4.9) com 3, 5 e 10 objetivos. Para cada número de objetivos foi definido um valor específico da p-norma aplicada em cada função, a saber  $\lfloor \ln(M)/\ln(2) \rfloor$  sendo  $M$  o número de objetivos do problema. Desta forma as Fronteiras Pareto deste problema estão localizadas em superfícies de p-norma constante com  $p$  igual a 2, 3 e 4 para os problemas com 3, 5 e 10 objetivos respectivamente. A convergência da população é controlada pela função  $g(\mathbf{x})$  descrita na equação (4.5). Em todos os experimentos foi utilizada uma população de 300 indivíduos. Cada problema tem  $M + 9$  variáveis e o número máximo de gerações é de  $300 \times M$ , sendo  $M$  o número de objetivos.

A qualidade das soluções obtidas foi avaliada pelas métricas de Hiper-Volume (HV), *Inverted Generational Distance* (IGD) e a Métrica de Distância Radial (*Radial Distance Metric* - RDM) proposta no capítulo 4 e estes valores estão reunidos na tabela 5. Os resultados de cada algoritmo/técnica foram comparados utilizando o teste de hipóteses não paramétrico de Wilcoxon [158, 109], considerando como hipótese nula  $H_0$  a inexistência de diferença entre a distribuição dos resultados.

A tabela 5 apresenta o valor médio e a variância das métricas Hiper-Volume HV, *Inverted Generational Distance* IGD e a Métrica de Distância Radial RDM obtidas em 30 instâncias dos algoritmos MOEA/D e NSGAIII utilizando vetores de peso gerados pelo método de Scheffé (indicado por MOEA/D e NSGAIII) e pelo método proposto (indicado por wv-MOEA/D e wv-NSGAIII). Para a métrica de HV o ponto de referência utilizado foi o ponto Nadir da Fronteira Pareto do problema, a saber  $N = (1, \dots, 1)$ . Para a métrica IGD foi gerada uma amostra da Fronteira Pareto com 100.000 pontos gerados pelo método de Scheffé.

Para o número de objetivos analisados os valores das métricas HV e IGD são muito próximos. Uma vez que estas métricas avaliam a convergência e a dispersão da população estes resultados indicam que as soluções obtidas são similares, mas estatisticamente diferentes com qualquer nível de significância  $\alpha$ . A tabela 6 apresenta os p-valores do teste não paramétrico de Wilcoxon comparando as métricas HV, IGD e RDM obtidas nas 30 instâncias de testes realizadas. Este teste procura verificar se existe alguma diferença estatisticamente significativa nos resultados obtidos em um mesmo algoritmo utilizando

diferentes metodologias de geração de vetores de peso. Os p-valores encontrados indicam que com qualquer nível de significância considerado não se pode afirmar em uma equivalência entre os métodos e os valores médios apresentados na tabela 5 indicam a superioridade do método Scheffé de geração de vetores de peso nas métricas utilizadas.

A tabela 7 apresenta os p-valores do teste não paramétrico de Wilcoxon comparando as métricas HV, IGD e RDM obtidas nas 30 instâncias de testes realizadas. Este teste procura verificar se existe alguma diferença estatisticamente significativa nos resultados obtidos em um mesmo método de geração de vetores de peso utilizando diferentes algoritmos. Os p-valores encontrados indicam que com qualquer nível de significância considerado não se pode afirmar em uma equivalência entre os algoritmos e os valores médios apresentados na tabela 5 indicam a superioridade do algoritmo NSGAIII segundo as métricas utilizadas. Os p-valores idênticos obtidos para o método de Scheffé em problemas de 3 e 5 objetivos se deve à homogeneidade dos resultados obtidos pelo algoritmo NSGAIII nestas instâncias como pode ser verificado pela ordem de grandeza das variâncias apresentada na tabela 5.

As figuras 44 a 48 ilustram os resultados obtidos em instâncias aleatórias dos algoritmos MOEA/D e NSGAIII utilizando vetores de peso gerados pelo método de Scheffé e pelo método proposto no problema pdMOP02 com 3, 5 e 10 objetivos. Uma vez que os algoritmos que utilizam a estratégia de decomposição/agregação distribuem as soluções obtidas alinhadas aos vetores de peso, as soluções obtidas pelos algoritmos que utilizam o método de Scheffé apresentam uma distribuição regular bem característica, dispostas em um padrão do tipo “grade”. Esta distribuição é ilustrada nas figuras 44a, 44c, 45a, 45c, 47a, 46a, 46c e 48a. Entretanto este padrão na distribuição dos pontos tem um efeito danoso, que é a presença de regiões do espaço vazias de soluções como pode ser observado

Objetivos	Algoritmo	HV	IGD	RDM
3	wvMOEAD	7.641e-01 (2.489e-05)	2.900e-02 (1.098e-07)	2.860e-05 (2.989e-10)
	MOEAD	7.706e-01 (3.615e-06)	2.901e-02 (2.184e-08)	1.192e-05 (1.084e-10)
	wvNSGAIII	7.633e-01 (1.886e-06)	2.842e-02 (3.144e-08)	5.493e-07 (1.800e-12)
	NSGAIII	7.739e-01 (2.432e-11)	2.866e-02 (5.884e-12)	1.127e-10 (3.555e-21)
5	wvMOEAD	1.060e+00 (1.932e-05)	1.753e-01 (3.890e-07)	7.630e-05 (9.094e-10)
	MOEAD	1.078e+00 (8.007e-06)	1.713e-01 (5.347e-09)	1.348e-05 (2.286e-10)
	wvNSGAIII	1.057e+00 (4.923e-06)	1.717e-01 (3.877e-07)	8.354e-06 (3.446e-10)
	NSGAIII	1.056e+00 (4.256e-07)	1.968e-01 (8.267e-12)	5.539e-09 (5.131e-16)
10	wvMOEAD	2.054e+00 (1.634e-04)	5.037e-01 (1.640e-06)	2.762e-04 (7.820e-09)
	MOEAD	2.083e+00 (8.077e-04)	4.963e-01 (9.851e-07)	1.738e-04 (4.261e-08)
	wvNSGAIII	1.983e+00 (1.178e-02)	5.182e-01 (2.326e-03)	3.513e-04 (6.182e-08)
	NSGAIII	2.059e+00 (7.237e-03)	5.345e-01 (4.361e-03)	6.376e-05 (1.029e-08)

Tabela 5 – Valores médios e variância das métricas Hiper-Volume (HV), *Inverted Generational Distance* (IGD) e a Métrica de Distância Radial (RDM) obtidas em 30 instâncias dos algoritmos MOEA/D e NSGAIII utilizando vetores de peso gerados pelo método de Scheffé (MOEA/D e NSGAIII) e pelo método proposto (wv-MOEAD e wv-NSGAIII)

Objetivos	Algoritmo	HV	IGD	RDM
3	MOEAD	2.668e-09	2.009e-01	2.878e-06
	NSGAIII	3.020e-11	8.485e-09	3.020e-11
5	MOEAD	3.020e-11	3.018e-11	2.370e-10
	NSGAIII	2.439e-03	3.020e-11	3.020e-11
10	MOEAD	7.043e-07	3.018e-11	1.336e-05
	NSGAIII	1.407e-04	1.188e-01	5.967e-09

Tabela 6 – p-valores para o teste de Wilcoxon pareado para as métricas utilizadas nos algoritmos MOEA/D e NSGAIII com as diferentes metodologias de geração de vetores de peso

nos gráficos de mapeamento angular do CAP-vis. Este fenômeno é mais evidente quando observado nos gráficos de coordenadas paralelas. Em particular, na figura 48a observa-se uma ausência de soluções no intervalo  $0.2 \leq x_i \leq 0.7$  em todos os objetivos. Tal fenômeno não ocorre quando o conjunto de vetores de peso é constituído de pontos quase igualmente espaçados como pode ser observado nas figuras 47b e 48b e no mapeamento angular do gráfico CAP-vis dos problemas que utilizaram o método apresentado neste trabalho para a geração de vetores de peso.

A falta de diversidade de soluções pode ser contornada com o aumento da população, mas em problemas com muitos objetivos o crescimento da população não é regular uma vez que esta apresenta um crescimento combinatório e desta forma seu tamanho pode se tornar tão grande que inviabilize o processo de otimização em um tempo razoável. Este problema não está presente no método de geração de vetores de peso proposto neste trabalho, uma vez que o tamanho da população pode ser definido sem restrições.

Apesar da inferioridade estatística observada na comparação das métricas de avaliação das soluções utilizadas, as soluções obtidas utilizando o método de geração de vetores de peso proposto apresentam boas medidas de convergência e dispersão das soluções,

Objetivos	Algoritmo	HV	IGD	RDM
3	Scheffé	3.018e-11	3.018e-11	3.018e-11
	WV	1.273e-02	4.616e-10	3.020e-11
5	Scheffé	3.020e-11	3.018e-11	3.018e-11
	WV	4.637e-03	3.020e-11	3.474e-10
10	Scheffé	2.052e-03	4.111e-07	3.563e-04
	WV	5.573e-10	1.442e-03	5.592e-01

Tabela 7 – p-valores para o teste de Wilcoxon pareado para as métricas utilizadas nos algoritmos MOEA/D e NSGAIII com as diferentes metodologias de geração de vetores de peso

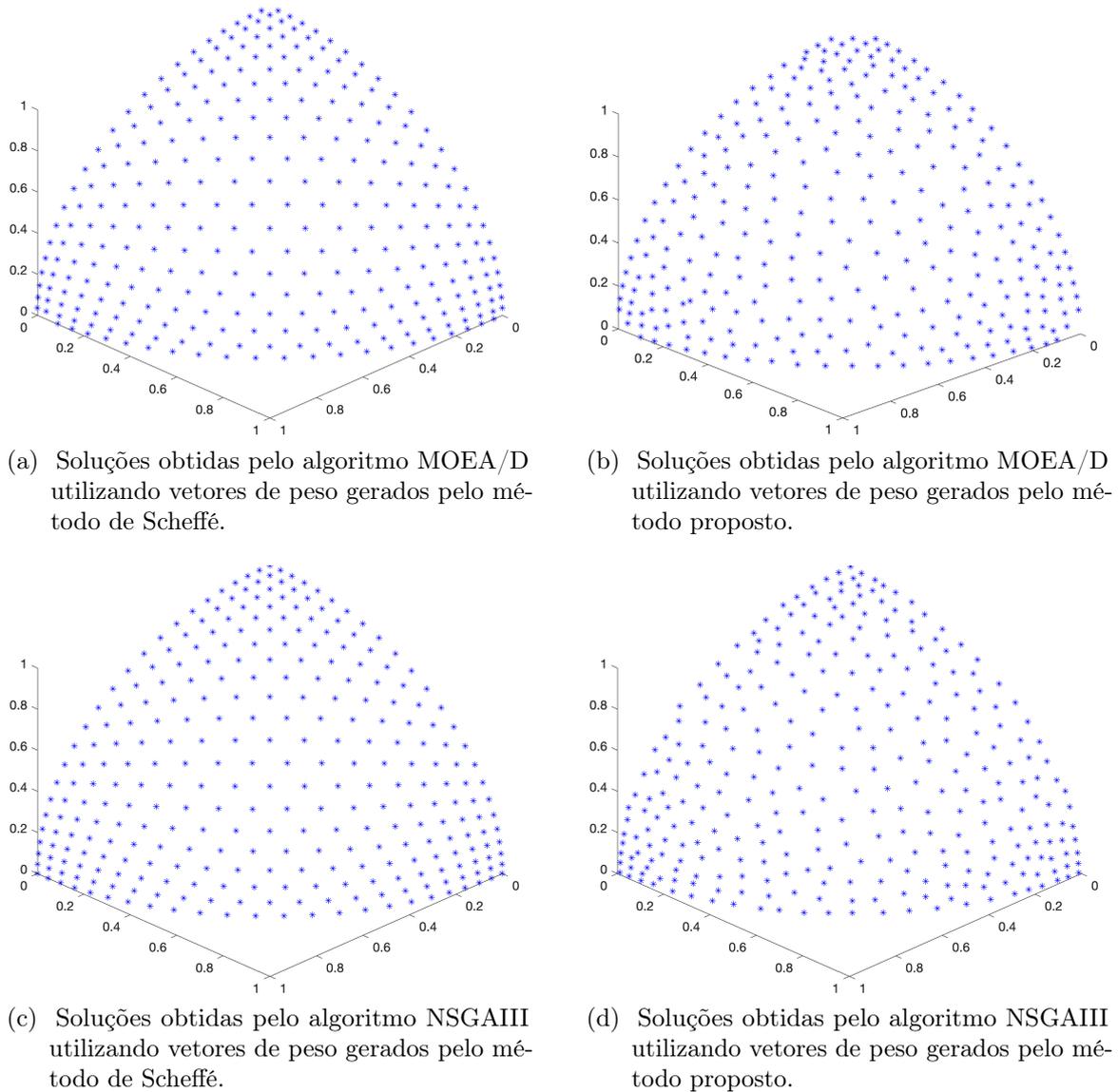
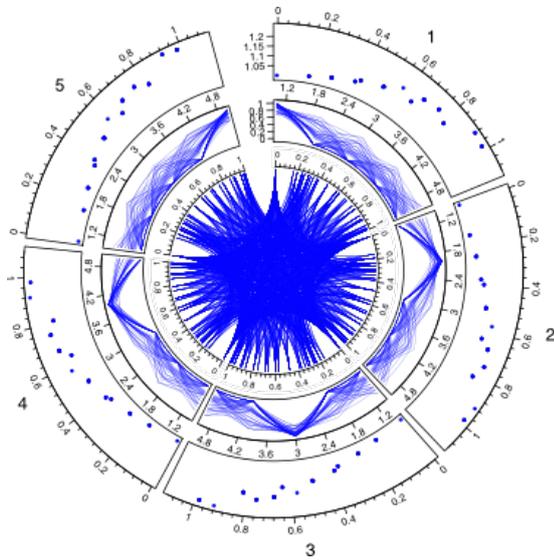


Figura 44 – Soluções obtidas pelo algoritmo MOEA/D e NSGAIII para o problema pd-MOP02 com valor de p-norma igual a 2 e 3 objetivos.

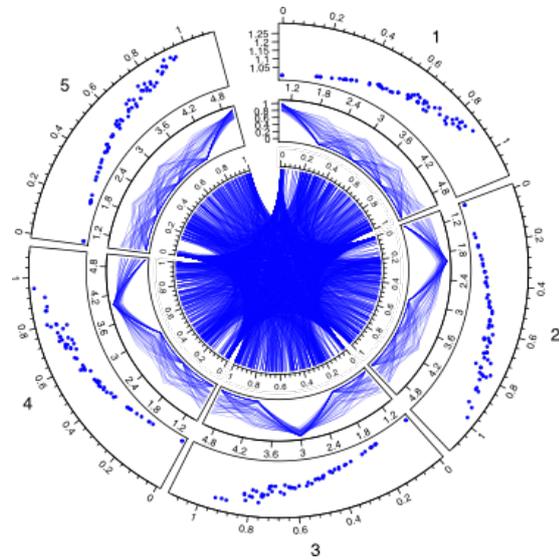
além de apresentarem uma diversidade de soluções muito maior nos casos analisados e de não apresentar limitações de tamanho de população, o que faz deste método uma alternativa viável e vantajosa ao método de Scheffé de geração de vetores de peso, principalmente em problemas com muitos objetivos.

## 6.4 Considerações finais

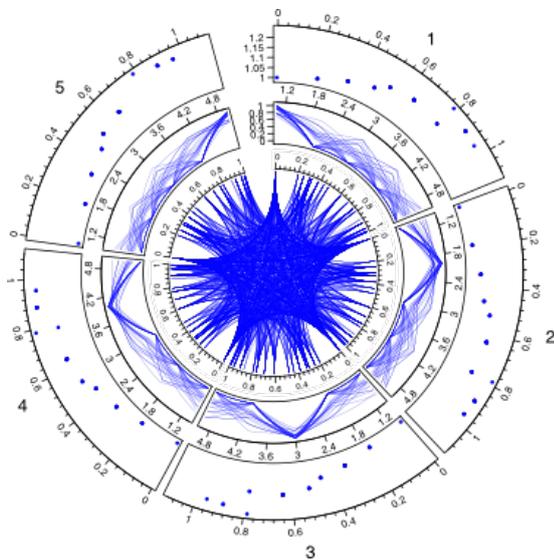
Os algoritmos de otimização multiobjetivo que utilizam a técnica de decomposição/agregação são indiscutivelmente superiores aos métodos que utilizam o critério de dominância de Pareto como instrumento de atualização da população entre iterações seguidas em problemas com mais do que dois objetivos. Entretanto a geração dos vetores de peso é limitante da aplicação desta classe de algoritmos uma vez que o número de



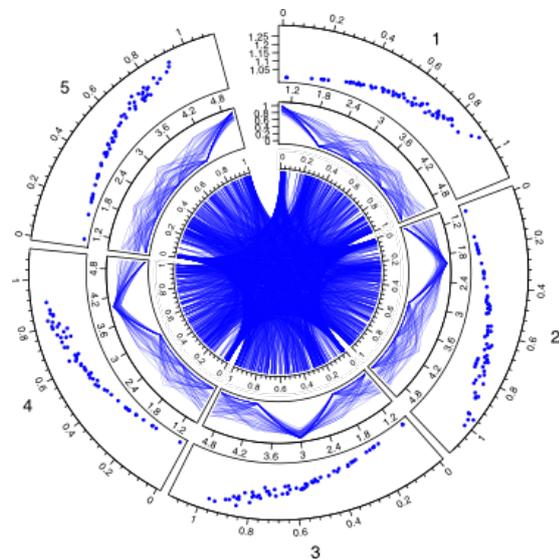
(a) Soluções obtidas pelo algoritmo MOEA/D utilizando vetores de peso gerados pelo método de Scheffé.



(b) Soluções obtidas pelo algoritmo MOEA/D utilizando vetores de peso gerados pelo método proposto.

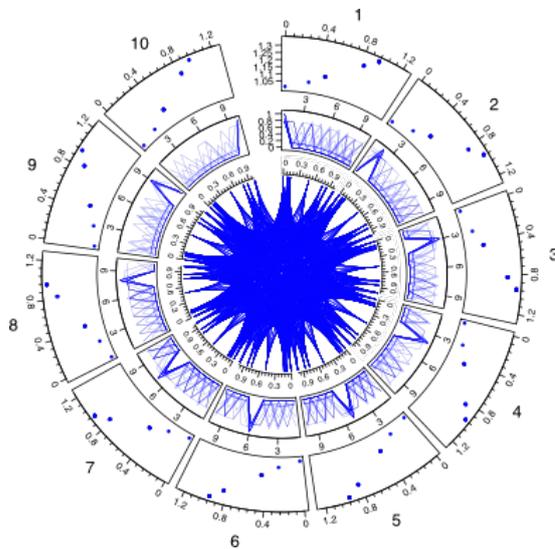


(c) Soluções obtidas pelo algoritmo NSGAIII utilizando vetores de peso gerados pelo método de Scheffé.

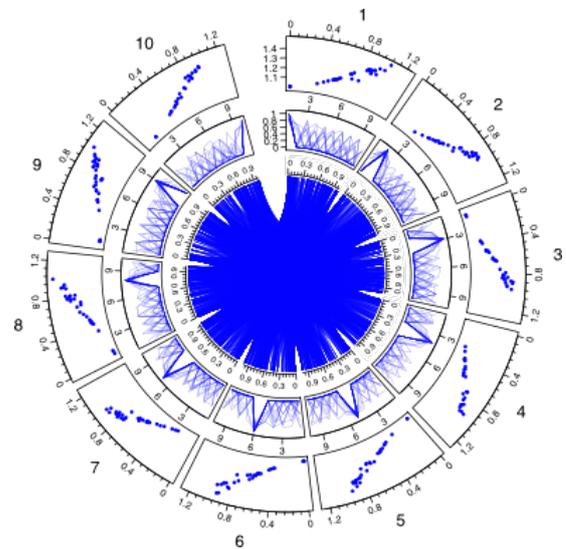


(d) Soluções obtidas pelo algoritmo NSGAIII utilizando vetores de peso gerados pelo método proposto.

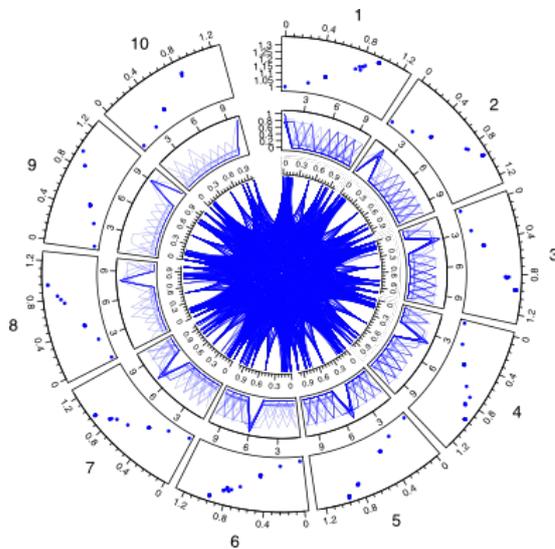
Figura 45 – Soluções obtidas pelo algoritmo MOEA/D e NSGAIII para o problema pd-MOP02 com 5 objetivos



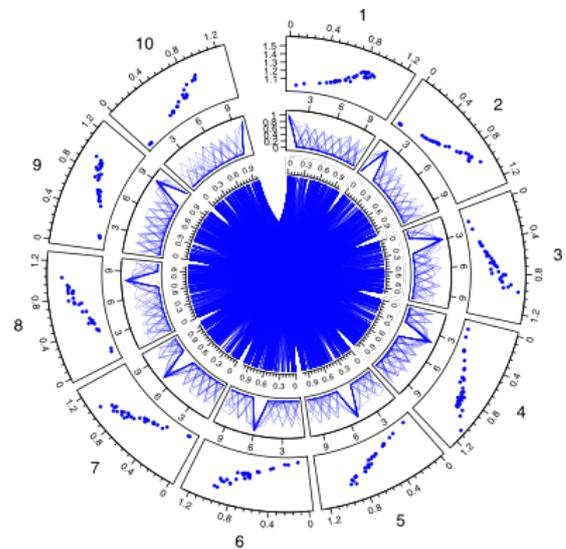
(a) Soluções obtidas pelo algoritmo MOEA/D utilizando vetores de peso gerados pelo método de Scheffé.



(b) Soluções obtidas pelo algoritmo MOEA/D utilizando vetores de peso gerados pelo método proposto.



(c) Soluções obtidas pelo algoritmo NSGAIII utilizando vetores de peso gerados pelo método de Scheffé.



(d) Soluções obtidas pelo algoritmo NSGAIII utilizando vetores de peso gerados pelo método proposto.

Figura 46 – Soluções obtidas pelo algoritmo MOEA/D para o problema pdMOP02 com 10 objetivos

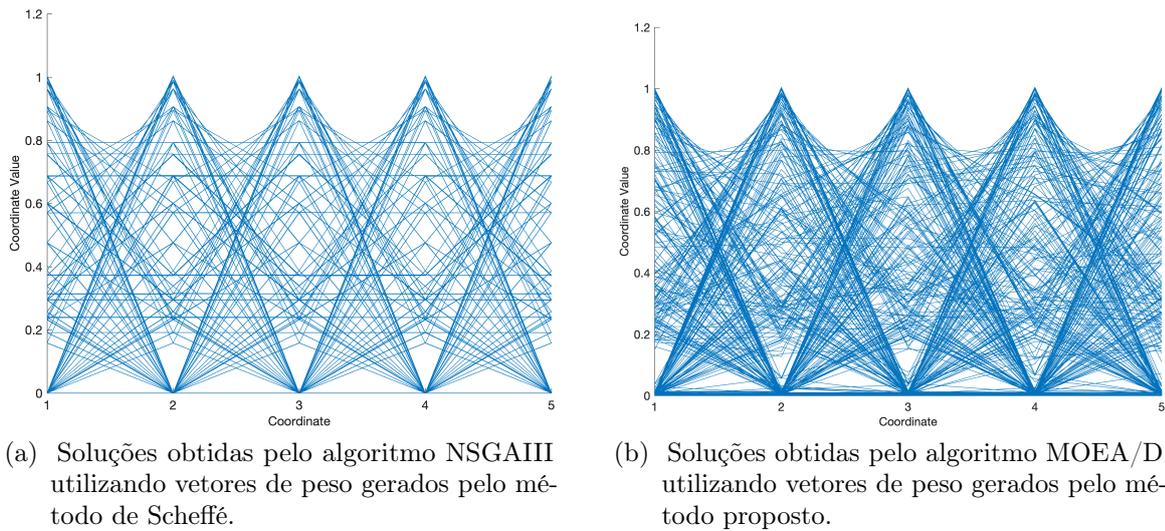


Figura 47 – Coordenadas paralelas das soluções obtidas pelo algoritmo NSGAIII para o problema pdMOP02 com 5 objetivos

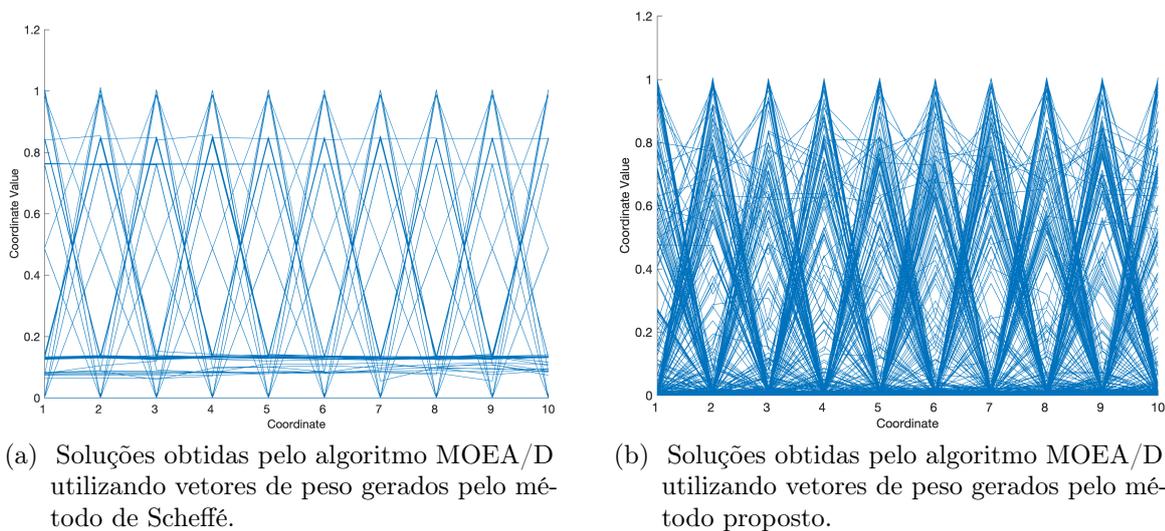


Figura 48 – Coordenadas paralelas das soluções obtidas pelo algoritmo MOEA/D para o problema pdMOP02 com 5 objetivos

vetores de peso cresce de forma binomial com o número de objetivos o que provoca um crescimento indesejado e desnecessário do tamanho da população ou a não representação de algumas regiões do espaço dos objetivos com a limitação do número de vetores de peso. Muito esforço por parte da comunidade científica é empregado na superação deste dilema e a proposta de geração de vetores de peso e de cones de vetores de peso apresentada neste capítulo faz parte deste movimento.

Os experimentos realizados mostraram que o método proposto apresenta bons resultados e é uma alternativa viável para a geração de vetores de peso no primeiro ortante do espaço dos objetivos. Além disto, a geração de cone de vetores de peso é um método sem precedentes na literatura e será determinante na exploração de regiões de interesse do

espaço dos objetivos. Este tema será tratado no capítulo 7.

Os métodos de geração de vetores de peso apresentados neste capítulo foram implementados em MATLAB e cópias destes algoritmos (*Weight Vector Generator* – WVG.m e *cone-Weight Vector Generator* – coneWVG.m) estão disponíveis para *download* no site do MINDS (<http://minds.eng.ufmg.br/>) e no github (<https://github.com/ivanreinaldo/WVG>)



## 7 Incorporação de região de interesse em problemas de otimização multiobjetivo

### 7.1 Introdução

Um algoritmo de otimização multiobjetivo (MOEA) genérico apresenta como solução de um problema de otimização um conjunto de pontos no espaço de decisão cuja imagem no espaço dos objetivos consiste em uma aproximação de toda a Fronteira Pareto do problema. Uma das características esperadas deste conjunto solução é a sua distribuição na Fronteira Pareto do problema que deve ser uniforme e conter pontos em toda a superfície. Um conjunto de soluções que apresente um agrupamento de pontos em uma região e poucas soluções em outra parte da Fronteira Pareto é claramente inferior a um conjunto de soluções com uma distribuição de pontos equilibrada em toda a extensão da Fronteira Pareto.

Encontrar um conjunto de soluções bem distribuído e suficientemente próximo da Fronteira Pareto de um problema de otimização multiobjetivo é uma tarefa difícil e em alguns casos ineficaz, como por exemplo as soluções extremas localizadas no bordo da Fronteira Pareto citadas no capítulo anterior. Um outro desafio presente em problemas multiobjetivo é a escolha de uma solução em particular entre as várias alternativas oferecidas pelo otimizador.

Por este motivo muitos algoritmos de otimização multi-objetivo baseados em preferências são apresentados na literatura especializada [108, 127, 90, 161, 154, 137]. Estes algoritmos são projetados para obter soluções localizadas em uma região específica do espaço dos objetivos, usualmente chamada de Região de Interesse (ROI). Este tipo de MOEA apresenta uma estratégia intermediária em otimização multiobjetivo: alguma informação *a priori* é necessária para definir a ROI e após a determinação das soluções pelo otimizador o tomador de decisão seleciona a solução satisfatória *a posteriori* ou reinicia o procedimento de otimização ajustado a ROI em um processo iterativo. Uma revisão atual do estado da arte da incorporação de preferências do decisor em otimização evolutiva multiobjetivo é apresentada por Bechikh et al. [9].

A determinação de uma ROI *a priori* faz uso do conhecimento prévio do problema por parte do tomador de decisão que pode desta forma definir um conjunto de preferências para as soluções a serem obtidas pelo otimizador, reduzindo o custo computacional do processo de otimização e o tempo empregado na escolha do conjunto de soluções finais.

A maneira de como o tomador de decisão expressa suas preferências em um problema

de otimização pode ser feito de três maneiras [103, 49]:

**Abordagem *a priori*:** As preferências do decisor são definidas antes do processo de otimização [108, 90, 49].

**Abordagem *a posteriori*:** O decisor é consultado após o processo de otimização, onde lhe é apresentado todo o conjunto de soluções [15].

**Abordagem interativa:** Ocorre uma ou mais consultas ao decisor durante o processo de otimização [14, 104].

Uma estratégia de incorporação das preferências do decisor no processo de otimização muito utilizada é o método Achievement Scalarizing Functions (ASF) [157, 103, 113]. Em uma ASF o tomador de decisão especifica um ponto de referência  $\mathbf{P}$  e o otimizador privilegia a busca por soluções próximas a este ponto. Existem diversas formas de se definir uma ASF, inclusive pela definição de novas relações de dominância, tais como a  $g$ -dominância [108] ou a  $r$ -dominância [128]. Os algoritmos R-NSGAI [33], *Light beam search* [32], dentre outros [143, 49] utilizam esta técnica. Uma deficiência deste método é a falta de precisão na determinação da ROI, que é definida pela proximidade entre a população e um ponto de referência  $\mathbf{P}$ . Desta forma a extensão da ROI depende da posição relativa entre o ponto  $\mathbf{P}$  e a Fronteira Pareto do problema.

Uma outra forma de incorporação *a priori* das preferências do decisor é por meio dos vetores de peso. Mohammadi et al. [107] utilizam uma ASF para atualizar os vetores de peso seguindo as preferências do decisor. Liu et al. [89] apresentam uma variação do algoritmo MOEA/D e definem as preferências do decisor por meio de um ou mais vetores de peso, gerados de forma usual e que são atualizados durante o processo evolutivo. Durante este processo, consultas são realizadas ao tomador de decisão que verifica a qualidade das soluções correntes e as orienta segundo um conjunto de preferências.

Cheng et al. [16, 18] apresentam o *Reference Vector Guided Evolutionary Algorithm* (RVEA). Este algoritmo utiliza a estratégia de decomposição/agregação. A função de agregação utiliza a distância angular penalizada (APD) e os  $N$  vetores de peso são atualizados a cada iteração  $t$  por meio da transformação

$$\mathbf{v}_{t+1,i} = \frac{\mathbf{v}_{0,i} \circ (\mathbf{z}_{t+1}^{max} - \mathbf{z}_{t+1}^{min})}{\|\mathbf{v}_{0,i} \circ (\mathbf{z}_{t+1}^{max} - \mathbf{z}_{t+1}^{min})\|} \quad (7.1)$$

sendo  $1 \leq i \leq N$ ,  $\mathbf{z}_{t+1}^{min}$  e  $\mathbf{z}_{t+1}^{max}$  os vetores com os valores mínimos e máximos de cada objetivo na iteração  $t+1$ ,  $\mathbf{v}_{0,i}$  os vetores de peso na primeira iteração (gerados pelo método de Scheffé) e o operador  $\circ$  sendo o produto de Hadamard <sup>1</sup>

<sup>1</sup> Se  $A = [a_{i,i}]$  e  $B = [b_{i,i}]$  são duas matrizes de mesma ordem então  $A \circ B = C = [c_{i,i}]$  com  $c_{i,i} = a_{i,i} \cdot b_{i,i}$ .

A articulação das preferências do decisor é feita pela definição de um vetor  $\mathbf{v}_c$  e um fator de escala  $r$  com  $1 < r < 1$ , utilizando a transformação

$$\mathbf{v}'_i = \frac{r\mathbf{v}_i + (1-r)\mathbf{v}_c}{\|r\mathbf{v}_i + (1-r)\mathbf{v}_c\|} \quad (7.2)$$

sendo  $\mathbf{v}'_i$  o vetor do novo conjunto de vetores de peso,  $\mathbf{v}_i$  um vetor do conjunto de vetores de peso original gerados pelo método de Scheffé e  $\mathbf{v}_c$  o vetor de preferências definido pelo tomador de decisão. Os vetores gerados pelas transformações (7.1) e (7.2) têm  $p$ -norma constante igual a um, com  $p = 2$ , ou seja, estão contidos em uma esfera de raio 1 no espaço dos objetivos.

A combinação convexa definida pela equação (7.2) é uma contração. Portanto o vetor  $\mathbf{v}_c$  é um ponto fixo e o novo conjunto de vetores de peso não é simétrico em relação a este vetor, que mantém sua posição relativa no novo conjunto de vetores. Desta forma, se  $\mathbf{v}_c$  está próximo do bordo do espaço dos objetivos então  $\mathbf{v}_c$  estará próximo do bordo do conjunto de vetores transformados  $\mathbf{v}'_i$ . Este desequilíbrio reduz a diversidade das soluções obtidas pois se espera uma simetria entre o vetor de preferências e o conjunto de soluções apresentadas pelo otimizador.

## 7.2 Método Proposto

Esta seção apresenta uma metodologia de incorporação de preferências do decisor em algoritmos de otimização multiobjetivo que utilizam a técnica da decomposição/agregação. A metodologia proposta define a ROI como sendo a interseção entre a Fronteira Pareto do problema e um cone de vetores. O conjunto de preferências do decisor é representado pelo vetor  $\mathbf{v}$  que define o eixo do cone e o ângulo  $\theta$  entre o vetor  $\mathbf{v}$  e a geratriz do cone define a extensão da região de interesse.

Uma vez que as funções de agregação utilizam a informação da posição relativa de um ponto no espaço dos objetivos para sua avaliação, além do cone de vetores  $\mathbf{C}$  que define a ROI é necessária a criação de cones de vetores auxiliares  $\mathbf{C}_i$ . Em cada vetor da base canônica  $\mathbf{e}_i$  é criado um cone auxiliar  $\mathbf{C}_i$  tendo este vetor como eixo do cone. Os vetores destes cones auxiliares são restritos ao primeiro ortante do espaço  $\mathbb{R}^M$ . Cada cone  $\mathbf{C}_i$  tem uma quantidade reduzida de vetores e um pequeno ângulo de abertura  $\theta$ . Por exemplo, se esta metodologia for aplicada no algoritmo MOEA/D, sugerimos que cada cone auxiliar  $\mathbf{C}_i$  tenha o mesmo tamanho da vizinhança  $B(i)$  definida neste algoritmo.

A metodologia de exploração de uma ROI por meio de um algoritmo evolutivo que utilize a técnica de decomposição/agregação pode ser resumida da seguinte forma:

- Defina a ROI utilizando um cone de vetores  $\mathbf{C}$ . Nesta etapa o tomador de decisão

define um vetor de preferências  $\mathbf{v}$  que por sua vez define o eixo do cone de vetores de preferência. Este vetor pode ser definido de várias maneiras:

1. As componentes deste vetor podem refletir o valor nominal desejado pelo tomador de decisão para cada um dos objetivos;
2. O tomador de decisão pode estabelecer um vetor de pesos para cada objetivo. Por exemplo, em um problema com cinco objetivos O primeiro o e terceiro objetivos podem ter o dobro de importância que os demais. Neste caso o vetor que define as preferências do decisor é  $\mathbf{v} = (2, 1, 2, 1, 1)$ .
3. O decisor pode utilizar alguma técnica de atribuição de pesos para os objetivos, tais como a SWARA [72]

O ângulo  $\theta$  de abertura deste cone define a diversidade das soluções a serem obtidas: se por um lado um valor pequeno de  $\theta$  torna a escolha da solução muito restrita, um valor grande de  $\theta$  definirá uma diversidade de soluções muito grande, que contradiz o conceito de Região de Interesse.

- Escolha o algoritmo evolutivo. Uma vez que esta técnica consiste na substituição do conjunto de vetores de peso original pelo cone de vetores  $\mathbf{C}$  e os cones auxiliares  $\mathbf{C}_i$ ,  $1 \leq i \leq M$ , qualquer algoritmo evolutivo que utilize a técnica de decomposição/agregação pode ser utilizado. O número de vetores nos cones auxiliares pode variar entre diferentes algoritmos.
- Substitua o conjunto de vetores de peso utilizado pelo algoritmo escolhido pelo cone de vetores  $\mathbf{C}$  e pelos cones auxiliares  $\mathbf{C}_i$ ,  $1 \leq i \leq M$ . Nenhum outro ajuste no algoritmo é requerido.

Para ilustrar o uso desta técnica foi realizado um experimento considerando três vetores  $\mathbf{v}_1$ ,  $\mathbf{v}_2$  e  $\mathbf{v}_3$  representando preferências com diferentes características e três ângulos de abertura  $\theta_1$ ,  $\theta_2$  e  $\theta_3$  que irão representar diferentes extensões da ROI para cada vetor de preferência, obtendo desta forma nove tipos diferentes de cones de vetores. As especificações destes vetores e ângulos são apresentadas na tabela 8

Vetor	Ângulo de abertura do cone
$\mathbf{v}_1 = (1, \dots, 1)$	$\theta_1 = 0.05 \times \Theta$
$\mathbf{v}_2 = (3, 1, \dots, 1)$	$\theta_2 = 0.15 \times \Theta$
$\mathbf{v}_3 = (1, 3, \dots, 3)$	$\theta_3 = 0.25 \times \Theta$

Tabela 8 – Características dos eixos  $\mathbf{v}_i$  e ângulos de abertura  $\theta_j$  dos cones de preferências utilizados nos experimentos

O vetor  $\mathbf{v}_1$  corresponde à hiper-diagonal do primeiro ortante do espaço dos objetivos. Para um problema onde  $[0, 1]$  é o menor intervalo de variação dos valores das soluções de

cada objetivo, este vetor representa o desejo do decisor de obter soluções equilibradas, onde nenhum objetivo tem um valor extremo, próximo de zero ou de um. Se o problema tem objetivos dissimilares, sua normalização neste intervalo faz com que este vetor represente soluções com as mesmas características, porém considerando o intervalo de variação de cada objetivo. O vetor  $\mathbf{v}_2$  corresponde a um conjunto de preferências onde o primeiro objetivo tem mais importância que os demais objetivos. Desta forma lhe foi atribuído o peso 3 enquanto que os demais têm peso 1. Esta situação corresponde ao desejo do tomador de decisão em encontrar soluções onde o primeiro objetivo tem um valor maior do que os demais. Geometricamente o cone de vetores ficará mais próximo do primeiro vetor  $\mathbf{e}_1$  da base canônica do espaço dos objetivos  $\mathbb{R}^M$ . No vetor  $\mathbf{v}_3$  esta situação se inverte: o primeiro objetivo tem menor importância que os demais. Geometricamente este cone de vetores ficará mais afastado do primeiro vetor  $\mathbf{e}_1$  da base canônica do espaço dos objetivos  $\mathbb{R}^M$ .

A escolha dos ângulos de abertura  $\theta_i$  tem como referência o ângulo  $\Theta$  entre a hiper-diagonal  $\mathbf{d} = (1, \dots, 1)$  do espaço dos objetivos e os vetores  $\mathbf{e}_i$  da base canônica do espaço dos objetivos  $\mathbb{R}^M$ , que vale  $\Theta = \arccos\left(\frac{1}{\sqrt{M}}\right)$ . Cada ângulo de abertura  $\theta_i$  é definido pelo produto deste ângulo  $\Theta$  por um fator de redução. Desta forma, o ângulo de abertura  $\theta_1$  irá definir um cone que “avista” uma região na Fronteira Pareto do problema cujo diâmetro é de aproximadamente 10% do ângulo entre a hiper-diagonal  $\mathbf{d}$  e algum dos eixos  $\mathbf{e}_i$  da base canônica do espaço dos objetivos  $\mathbb{R}^M$ . A mesma descrição vale para os outros ângulos  $\theta_2$  e  $\theta_3$ , porém com ângulo de visada iguais a 30% e 50% respectivamente. Esta escolha de ângulos irá gerar ROI's com diferentes características (restrita, intermediária e ampla respectivamente).

Os experimentos foram realizados em problemas com 3, 5 e 10 objetivos. Foi selecionado o problema pd-MOP02 com norma igual a  $\lceil \frac{\log(M)}{\log(2)} \rceil$ , sendo  $M$  o número de objetivos, por ser o problema proposto adequado a Otimização Multiobjetivo e com Muitos Objetivos.

Os resultados foram analisados sob dois aspectos: a pertinência das soluções na ROI, a convergência da população na Fronteira Pareto de cada ROI e diversidade. A pertinência das soluções na ROI foi medida pela contagem das soluções que atingiram a região de interesse definida e sua vizinhança. Foram consideradas duas vizinhanças para cada ROI. Cada vizinhança é definida da seguinte forma:

- Seja  $\mathbf{C}$  o cone definido pelo eixo  $\mathbf{v}_i$  e ângulo  $\theta_j$ ;
- Sejam  $\mathbf{C}_1$  e  $\mathbf{C}_2$  os cones definidos pelo eixo  $\mathbf{v}_i$  e ângulos  $\theta_a$  e  $\theta_b$  respectivamente, sendo

$$\theta_j < \theta_a \leq 1.1 \times \theta_j < \theta_b \leq 1.2 \times \theta_j$$

- A primeira vizinhança, denotada por  $R1$ , é o lugar geométrico dos pontos que estão no cone  $C_1$  mas não estão no cone  $C$ . A segunda vizinhança, denotada por  $R2$ , é o lugar geométrico dos pontos que estão no cone  $C_2$  mas não estão no cone  $C_1$  (e consequentemente não estão no cone  $C$ ).

Como a seção transversal do cone de vetores é um círculo, a sua interseção com a Fronteira Pareto será uma superfície cujo bordo é curva suave e fechada em um problema com três objetivos com Fronteira Pareto conexa. Nesta situação, as vizinhanças  $R1$  e  $R2$  correspondem a anéis concêntricos envolvendo esta superfície. A espessura de cada um destes anéis corresponde a um ângulo de visada de 10% do ângulo que define o cone de vetores.

A convergência dos pontos na ROI será medida pelas métricas IGD e RDM. A métrica IGD requer uma amostra da Fronteira Pareto e é sensível à qualidade desta amostra. Para produzir resultados de melhor qualidade, a amostra da Fronteira Pareto será feita por meio de um cone de vetores de peso com as mesmas configurações de cada ROI. Cada uma desta amostra foi composta por 800 vetores com a mesma norma dos pontos da Fronteira Pareto.

Nos experimentos foram utilizados os algoritmos NSGA-III e MOEA/D, sendo que neste último foram implementadas as melhorias descritas no capítulo 2. Em todas as simulações foi utilizada uma população de 300 indivíduos e consideradas  $1e5$  avaliações de função objetivo como critério de parada. Foram utilizados os operadores genéticos usuais (recombinação SBX e mutação polinomial) e a função PBI como técnica de agregação. Foram executadas 30 instâncias de teste para cada algoritmo. Foram considerados dois cenários: O primeiro deles considera o uso dos cones de vetores de peso no processo de decomposição/agregação. Os resultados obtidos estão agrupados nas linhas indicadas pelos algoritmos coneMOEA/D e coneNSGA-III. O segundo cenário considera uma seleção *a posteriori* das soluções em cada ROI e vizinhanças, obtidas por meio dos algoritmos MOEA/D e NSGA-III utilizando o conjunto de vetores de peso gerados pelo método usual (Scheffé).

As tabelas 9 a 11 apresentam os resultados obtidos. A coluna IGD e RDM apresentam o valor médio das métricas IGD e RDM das soluções obtidas na ROI nas 30 instâncias realizadas em cada cenário. A coluna #ROI mostra o valor médio do número de soluções que foram obtidas na ROI. As colunas RDM R1 e RDM R2 apresentam o valor médio da métrica RDM nas vizinhanças R1 e R2 e as colunas # R1 e # R2 indicam o valor médio do número de soluções que foram encontradas em cada uma destas vizinhanças. Se não foi encontrada solução em uma região específica, sua métrica será indicada por  $\star$  e sua contagem de pontos será igual a zero.

Os resultados obtidos mostram claramente a superioridade da abordagem *a priori*

MOEA	IGD	RDM	# ROI	RDM R1	# R1	RDM R2	#R2
<b>v1</b>							
<b>3 objetivos</b>							
coneMOEA/D	4.105e-03	3.222e-03	269	3.241e-03	31	*	0
coneNSGAIII	1.995e-03	4.275e-05	294	8.987e-05	5.2	1.835e-03	0.4
MOEA/D	*	*	0	*	0	*	0
NSGAIII	*	*	0	*	0	*	0
<b>5 objetivos</b>							
coneMOEA/D	2.410e-02	9.725e-03	268	9.733e-03	32.0	*	0
coneNSGAIII	2.041e-02	1.272e-04	292.9	1.871e-04	7.1	*	0
MOEA/D	*	*	0	*	0	*	0
NSGAIII	*	*	0	*	0	*	0
<b>10 objetivos</b>							
coneMOEA/D	7.158e-02	8.395e-03	211.2	8.415e-03	74.9	9.210e-03	12.3
coneNSGAIII	6.363e-02	2.996e-04	281.1	3.715e-04	8.9	*	0
MOEA/D	*	*	0	*	0	*	0
NSGAIII	*	*	0	*	0	*	0
<b>v2</b>							
<b>3 objetivos</b>							
coneMOEA/D	6.133e-03	5.384e-03	254.6	5.397e-03	27.6	5.403e-03	17.9
coneNSGAIII	1.969e-03	1.041e-05	296.6	4.114e-05	2.7	6.799e-06	0.2
MOEA/D	*	*	0	2.103e-03	3	*	0
NSGAIII	*	*	0	9.273e-06	2	*	0
<b>5 objetivos</b>							
coneMOEA/D	3.310e-02	9.450e-03	229.6	9.367e-03	54.2	9.287e-03	16.1
coneNSGAIII	2.825e-02	4.383e-05	295.6	8.701e-05	4.2	5.676e-03	0.07
MOEA/D	8.009e-02	1.994e-03	1	*	0	*	0
NSGAIII	*	*	0	*	0	*	0
<b>10 objetivos</b>							
coneMOEA/D	6.064e-02	7.030e-03	155.7	6.993e-03	84.3	6.936e-03	37.6
coneNSGAIII	5.115e-02	1.476e-04	235.7	3.051e-04	13.73	9.062e-04	0.6
MOEA/D	*	*	0	*	0	*	0
NSGAIII	*	*	0	*	0	*	0
<b>v3</b>							
<b>3 objetivos</b>							
coneMOEA/D	5.562e-03	4.818e-03	261.5	4.823e-03	26.8	4.825e-03	11.7
coneNSGAIII	1.973e-03	2.311e-05	295.7	6.040e-05	4.2	1.051e-03	0.07
MOEA/D	4.104e-02	2.002e-03	1	*	0	*	0
NSGAIII	3.739e-02	8.589e-07	1	*	0	*	0
<b>5 objetivos</b>							
coneMOEA/D	2.295e-02	9.008e-03	251.1	8.991e-03	48.7	6.956e-03	0.2
coneNSGAIII	1.957e-02	1.355e-04	292.4	1.757e-04	7.5	1.471e-03	0.07
MOEA/D	*	*	0	*	0	*	0
NSGAIII	*	*	0	*	0	*	0
<b>10 objetivos</b>							
coneMOEA/D	6.734e-02	8.501e-03	217.7	8.547e-03	65.2	8.676e-03	7.2
coneNSGAIII	6.255e-02	2.861e-04	251.8	3.828e-04	8.2	*	0
MOEA/D	*	*	0	*	0	*	0
NSGAIII	*	*	0	*	0	*	0

Tabela 9 – Valores médios das métricas IGD e RDM obtidas nos experimentos para os cones definidos pelo ângulo de abertura  $\theta_1$

MOEA	IGD	RDM	# ROI	RDM R1	# R1	RDM R2	# R2
<b>v1</b>							
<b>3 objetivos</b>							
coneMOEA/D	8.270e-03	4.307e-03	267.5	4.363e-03	32.5	*	0
coneNSGAIII	5.885e-03	5.353e-05	297.1	7.893e-05	2.6	4.414e-03	0.2
MOEA/D	4.446e-02	2.311e-03	6	*	0	2.353e-03	6
NSGAIII	4.420e-02	4.022e-07	6	*	0	1.884e-07	6
<b>5 objetivos</b>							
coneMOEA/D	4.949e-02	6.340e-03	262	6.350e-03	38.0	*	0
coneNSGAIII	4.694e-02	1.753e-04	295.9	3.398e-04	4.1	*	0
MOEA/D	*	*	0	*	0	*	0
NSGAIII	*	*	0	*	0	*	0
<b>10 objetivos</b>							
coneMOEA/D	1.979e-01	1.193e-02	197.2	1.194e-02	100	1.210e-02	2.8
coneNSGAIII	1.855e-01	3.951e-04	276	4.041e-04	3.967	*	0
MOEA/D	*	*	0	*	0	*	0
NSGAIII	*	*	0	*	0	*	0
<b>v2</b>							
<b>3 objetivos</b>							
coneMOEA/D	7.599e-03	3.585e-03	272.1	3.645e-03	27.9	*	0
coneNSGAIII	5.887e-03	2.165e-05	297.5	4.549e-05	2.5	*	0
MOEA/D	3.114e-02	2.062e-03	12	1.787e-03	2	*	0
NSGAIII	3.298e-02	2.747e-06	11	1.702e-07	3	*	0
<b>5 objetivos</b>							
coneMOEA/D	4.512e-02	5.620e-03	249.3	5.477e-03	50.7	*	0
coneNSGAIII	4.299e-02	8.405e-05	294.8	3.392e-04	5.133	3.501e-05	0.07
MOEA/D	1.589e-01	1.994e-03	1	*	0	*	0
NSGAIII	*	*	0	*	0	4.793e-06	1
<b>10 objetivos</b>							
coneMOEA/D	1.649e-01	8.580e-03	208.5	8.265e-03	86.53	8.226e-03	5
coneNSGAIII	1.721e-01	1.208e-03	205.3	1.603e-03	24.2	2.639e-03	1.8
MOEA/D	*	*	0	*	0	*	0
NSGAIII	*	*	0	*	0	*	0
<b>v3</b>							
<b>3 objetivos</b>							
coneMOEA/D	7.650e-03	3.802e-03	273.2	3.833e-03	26.8	*	0
coneNSGAIII	5.912e-03	5.716e-05	296.1	1.052e-04	3.9	*	0
MOEA/D	3.719e-02	2.078e-03	8	*	0	2.237e-03	2
NSGAIII	3.674e-02	1.388e-06	8	*	0	1.784e-06	5
<b>5 objetivos</b>							
coneMOEA/D	4.829e-02	5.680e-03	259.8	5.658e-03	40.2	*	0
coneNSGAIII	4.603e-02	1.752e-04	295.8	4.079e-04	4.2	*	0
MOEA/D	*	*	0	*	0	*	0
NSGAIII	*	*	0	*	0	*	0
<b>10 objetivos</b>							
coneMOEA/D	1.948e-01	1.077e-02	191.6	1.073e-02	105.7	1.050e-02	2.7
coneNSGAIII	1.831e-01	6.276e-04	251.5	7.276e-04	8.533	*	0
MOEA/D	*	*	0	*	0	*	0
NSGAIII	*	*	0	*	0	*	0

Tabela 10 – Valores médios das métricas IGD e RDM obtidas nos experimentos para os cones definidos pelo ângulo de abertura  $\theta_2$

MOEA	IGD	RDM	# ROI	RDM R1	# R1	RDM R2	# R2
<b>v1</b>							
<b>3 objetivos</b>							
coneMOEA/D	1.131e-02	3.232e-03	267.1	3.330e-03	32.9	*	0
coneNSGAI	9.718e-03	4.147e-05	297.6	5.495e-05	2.4	*	0
MOEA/D	4.120e-02	2.338e-03	18	2.353e-03	3	2.271e-03	6
NSGAI	4.106e-02	1.160e-06	18	5.163e-06	9	*	0
<b>5 objetivos</b>							
coneMOEA/D	8.055e-02	5.733e-03	260.3	5.693e-03	39.7	*	0
coneNSGAI	7.709e-02	3.920e-04	295.7	4.768e-04	4.3	*	0
MOEA/D	*	*	0	*	0	*	0
NSGAI	*	*	0	*	0	4.371e-06	5
<b>10 objetivos</b>							
coneMOEA/D	3.236e-01	1.409e-02	165.7	1.409e-02	129.3	1.382e-02	5
coneNSGAI	2.944e-01	6.320e-04	272.5	3.166e-03	7.6	3.395e-02	0.1
MOEA/D	*	*	0	*	0	*	0
NSGAI	*	*	0	*	0	*	0
<b>v2</b>							
<b>3 objetivos</b>							
coneMOEA/D	1.146e-02	3.943e-03	272.8	3.997e-03	27.2	*	0
coneNSGAI	9.912e-03	2.392e-05	298.2	6.253e-05	1.8	*	0
MOEA/D	3.103e-02	1.994e-03	30	1.687e-03	6	1.782e-03	6.1
NSGAI	3.129e-02	2.464e-06	29.4	7.157e-07	6.6	6.075e-06	8
<b>5 objetivos</b>							
coneMOEA/D	7.578e-02	4.718e-03	252	4.542e-03	48	*	0
coneNSGAI	7.325e-02	1.708e-04	291.4	4.470e-04	8.5	4.895e-04	0.1
MOEA/D	2.698e-01	1.994e-03	1	*	0	1.097e-03	4
NSGAI	3.341e-01	4.793e-06	1	1.184e-06	4	*	0
<b>10 objetivos</b>							
coneMOEA/D	2.849e-01	8.641e-03	191.1	8.283e-03	105.9	7.816e-03	2.9
coneNSGAI	3.072e-01	1.932e-03	214.3	2.237e-03	43.73	2.688e-03	13.4
MOEA/D	*	*	0	*	0	*	0
NSGAI	*	*	0	*	0	*	0
<b>v3</b>							
<b>3 objetivos</b>							
coneMOEA/D	1.128e-02	3.757e-03	277.4	3.768e-03	22.6	*	0
coneNSGAI	9.784e-03	3.273e-05	298.2	1.160e-04	1.8	*	0
MOEA/D	3.695e-02	2.003e-03	23	2.056e-03	8	1.809e-03	4
NSGAI	3.688e-02	2.116e-06	23	1.133e-06	8	1.688e-06	6
<b>5 objetivos</b>							
coneMOEA/D	7.786e-02	4.093e-03	266	3.937e-03	34	*	0
coneNSGAI	7.493e-02	1.353e-04	296.8	2.878e-04	3.2	*	0
MOEA/D	*	*	0	1.442e-03	4	*	0
NSGAI	*	*	0	*	0	*	0
<b>10 objetivos</b>							
coneMOEA/D	3.209e-01	1.315e-02	157.6	1.297e-02	137.4	1.262e-02	5
coneNSGAI	3.456e-01	5.483e-03	216.3	4.666e-03	7.2	4.645e-02	1.5
MOEA/D	*	*	0	*	0	*	0
NSGAI	*	*	0	*	0	*	0

Tabela 11 – Valores médios das métricas IGD e RDM obtidas nos experimentos para os cones definidos pelo ângulo de abertura  $\theta_3$

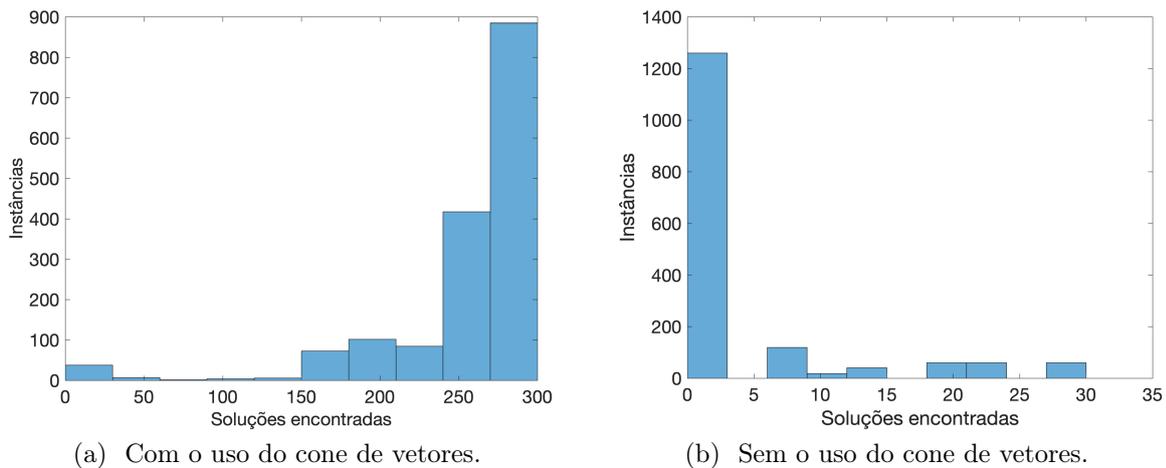


Figura 49 – Distribuição valor médio do número de soluções encontradas na ROI em todos os cenários

sobre a abordagem *a posteriori*. Dos 54 cenários analisados com o uso dos cones de vetores, 39 deles apresentam um valor médio de incidência de vetores na ROI definida superior a 250 pontos em uma população de 300 indivíduos enquanto que em apenas 6 cenários esta contagem é inferior a 200 observações, todas observados em problemas com 10 objetivos. Com a utilização do método de Scheffé para a geração de cones de vetores em apenas 18 cenários foi observada uma contagem de pontos na ROI acima de zero, sendo que nos 12 casos onde foi observado mais de um ponto, todos eles tratam de problemas com 3 objetivos. Além disto, considerando o cenário com o uso de cones de vetores, em 32 dos 54 cenários observados toda a população de 300 indivíduos ficou concentrada na região definida pela ROI e suas vizinhanças R1 e R2, sendo que em apenas 7 casos esta contagem foi inferior a 280 indivíduos, todos eles concentrados em problemas com 10 objetivos. O menor valor observado foi com 224 soluções nesta região definida pelo vetor  $\mathbf{v}_3$  com o ângulo  $\theta_3$ . Em apenas 23 cenários o método *a posteriori* encontrou alguma solução na ROI ou em suas vizinhanças R1 e R2, sendo que em apenas 11 casos este número foi superior a 10 observações, com um máximo de 44 observações na região definida pelo vetor  $\mathbf{v}_2$  com o ângulo  $\theta_3$  no problema com 3 objetivos. A figura 49 mostra o histograma da distribuição da frequência dos valores médios encontrados na ROI em todos os cenários com o uso do cone de vetores. Além disto não foi observada diferença significativa entre o tamanho da população encontrada nas diferentes regiões com a variação do ângulo de abertura dos cones que definem as ROI's.

Quando à convergência, em todos os cenários avaliados os valores da métrica RDM indicam uma boa aproximação das soluções obtidas em relação à Fronteira Pareto, sendo que estes valores são proporcionais ao número de objetivos: quanto menor o número de objetivos, menor o valor da métrica. Este padrão é previsível uma vez que a complexidade do problema cresce com a dimensão do espaço dos objetivos. Uma comparação entre os

algoritmos indica que o algoritmo MOEA/D, mesmo com os aperfeiçoamentos apresentados recentemente pela literatura especializada, apresenta um rendimento ligeiramente inferior quando comparada com o algoritmo NSGA-III padrão. Mais uma vez este comportamento é esperado uma vez que o algoritmo NSGA-III é mais recente (2014) do que o algoritmo MOEA/D (2007). A comparação entre as técnicas de geração de vetores de peso indica uma performance ligeiramente superior dos algoritmos que utilizam o cone de vetores em relação aos algoritmos que utilizam o método de Scheffé. Uma vez que no problema utilizado as variáveis do espaço de decisão responsáveis pela convergência e dispersão são independentes, este fenômeno pode ser explicado pelas características do método de agregação utilizado: O método de agregação PBI é definido pela expressão

$$d = d_1 + \alpha d_2 \quad (7.3)$$

sendo  $d_1$  a distância do ponto  $P$  ao ponto utópico  $Z$  e  $d_2$  a distância do ponto  $P$  ao vetor de peso  $\mathbf{w}$  utilizado, sendo  $\alpha$  um fator de penalidade. Nos experimentos realizados foi utilizado  $\alpha = 5$ . O algoritmo procura a todo instante fazer o ajuste dos pontos procurando um equilíbrio entre convergência e dispersão guiado pela equação (7.2). Nos cenários onde não está definida a ROI a densidade dos vetores de peso é menor e desta forma o valor de  $d_2$  na expressão (7.2) é em média maior. Desta forma são selecionados pontos com um melhor valor da variável  $d_2$  em detrimento do valor de  $d_1$ , o que sacrifica a convergência da população. Com o uso do cone de vetores de peso, a densidade dos vetores de peso é maior, o que reduz o valor de  $d_2$  na etapa final do processo evolutivo. Desta forma são escolhidos pontos cujo valor de  $d_1$  são melhores, o que favorece a convergência da população.

Os valores obtidos pela métrica IGD na ROI estão em harmonia com os valores da métrica RDM nos casos analisados. Como a métrica IGD é dependente e sensível de uma amostra da Fronteira Pareto, esta métrica só foi utilizada para medir a performance da população na ROI e não nas vizinhanças R1 e R2. Isto se deve ao formato característico destas vizinhanças: cada uma delas corresponde a uma região estreita no entorno da ROI, sem pontos na região central onde se localiza a mesma. Apesar de ser possível se produzir amostras da Fronteira Pareto com estas características, sua produção tem um elevado custo computacional o que desencoraja o seu uso.

A figura 50 exibe as soluções obtidas pelos algoritmos coneMOEA/D e coneNSGAIII em duas instâncias com três objetivos. A figura 50a ilustra uma solução obtida pelo algoritmo coneMOEA/D no problema pdMOP02 tendo a ROI definida pelo cone obtido pelo vetor  $\mathbf{v}_1$  e ângulo  $\theta_1$ . Nesta figura podemos observar os três cones auxiliares definidos pelos eixos  $e_1, e_2$  e  $e_3$  da base canônica. Devido ao alinhamento da ROI com o ângulo de visada utilizado no gráfico e sua pequena dimensão, uma ampliação das soluções obtidas na ROI é apresentada no canto superior direito desta figura. Nesta ampliação podemos observar com mais detalhes a posição relativa das soluções obtidas (pontos azuis) e a amostra da ROI (pontos pretos) que foi utilizada no cálculo da métrica IGD.

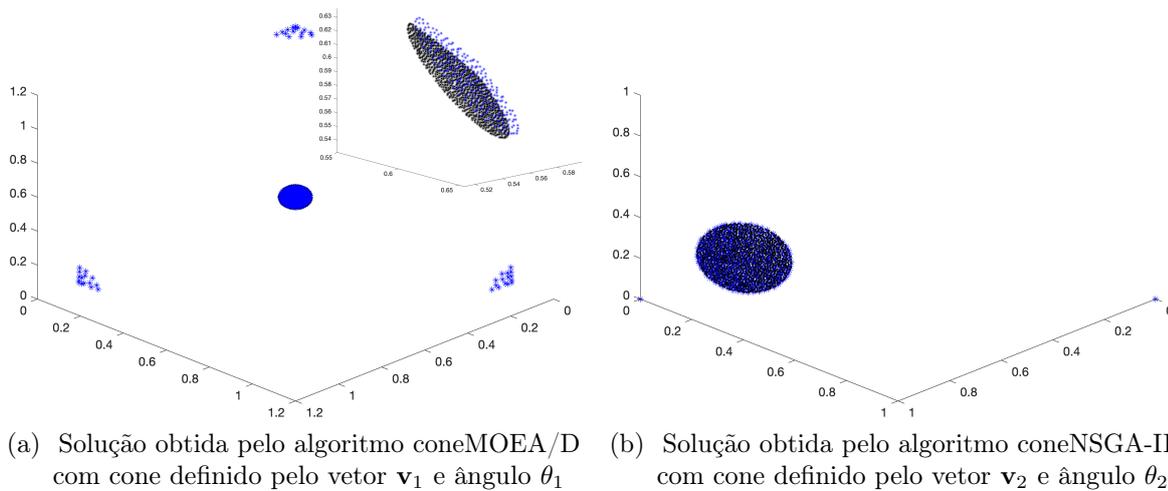


Figura 50 – Soluções obtidas pelos algoritmos coneMOEA/D e coneNSGA-III no problema pd-MOP02 com 3 objetivos.

A figura 50b ilustra uma solução obtida pelo algoritmo coneNSGAIII tendo a ROI definida pelo cone obtido pelo vetor  $\mathbf{v}_2$  e ângulo  $\theta_2$ . Nesta figura podemos observar os pontos localizados na extremidade dos vetores da base canônica utilizados para guiar o processo evolutivo em direção ao cone de preferências. As soluções obtidas são representadas por meio de pontos azuis e a amostra da ROI é representada por pontos pretos.

A figura 51 exibe as soluções obtidas pelos algoritmos coneMOEA/D e coneNSGAIII em duas instâncias com cinco objetivos utilizando o CAP-vis. A figura 51a ilustra uma solução obtida pelo algoritmo coneNSGAIII no cenário onde a ROI é definida pelo cone obtido pelo vetor  $\mathbf{v}_2$  e ângulo  $\theta_1$ . Em todos os setores e trilhas deste gráfico é possível ver o ponto auxiliar utilizado pelo algoritmo. Também é possível se observar que as soluções obtidas preservam as características impostas pelo vetor de preferências utilizado: todas as soluções estão próximas do vetor  $e_1$  da base canônica. Esta simetria em relação à importância dada aos demais objetivos reflete na localização das soluções nesta região do espaço: os pontos estão representados na região central do mapeamento angular. Como o ângulo  $\theta_1$  define uma ROI muito restrita, todas as soluções são representadas no setor 1 deste gráfico apresentam uma pequena extensão, o que indica que de fato todas as soluções estão concentradas nesta porção do espaço 5-dimensional. A trilha 2 do setor 1, que contém o gráfico em coordenadas paralelas, bem como o diagrama de cordas na região central do gráfico também confirmam o efeito do valor pequeno dado ao ângulo  $\theta_1$ , que faz com que os valores de cada coordenada apresentem uma pequena variação.

O vetor  $\mathbf{v}_3$  representa um conjunto de preferências onde o primeiro objetivo recebe um peso menor do que os demais, caracterizando uma situação oposta daquela representada pelo vetor  $\mathbf{v}_2$ . Esta situação pode ser observada na figura 51b, que ilustra uma solução obtida pelo algoritmo coneNSGAIII no cenário onde a ROI é definida pelo cone obtido

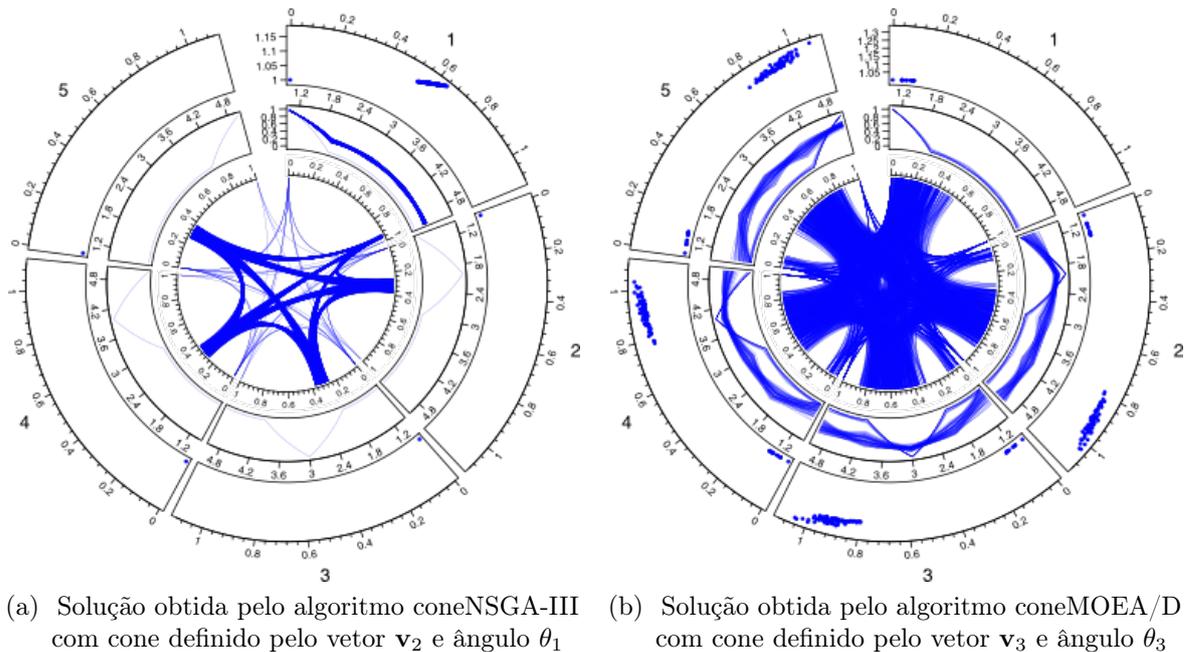
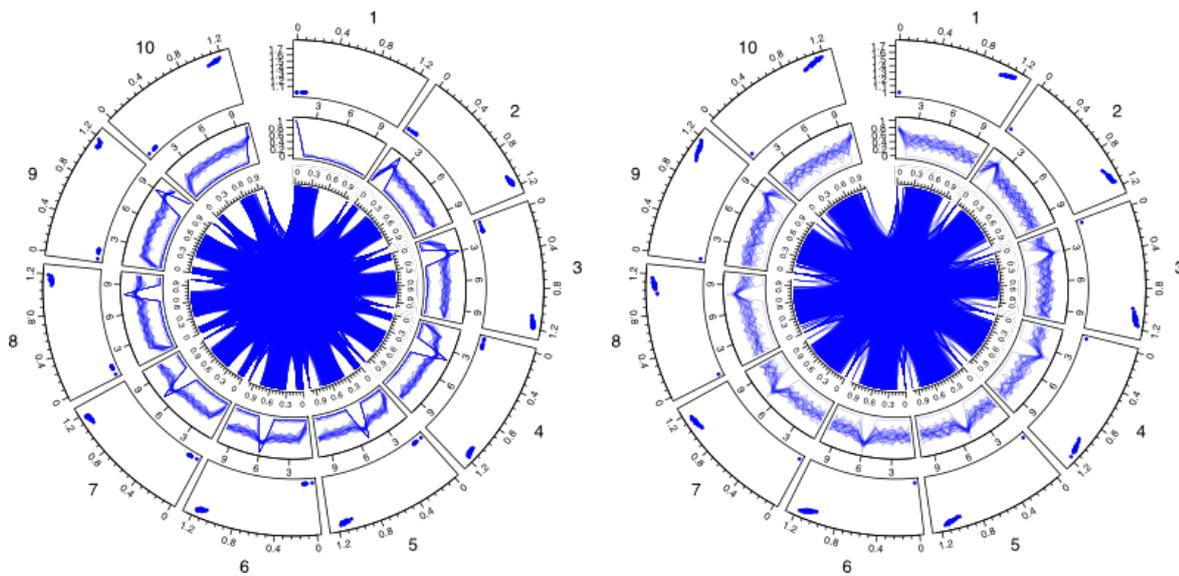


Figura 51 – Soluções obtidas pelos algoritmos coneMOEA/D e coneNSGA-III no problema pd-MOP02 com 5 objetivos.

pelo vetor  $\mathbf{v}_3$  e ângulo  $\theta_3$ . O antagonismo entre as preferências destes dois vetores pode ser visto quando comparamos as figuras 51a e 51b. Além disso, a abertura maior do cone definida pelo ângulo  $\theta_3$  também pode ser observada pela extensão do agrupamento dos pontos representados no mapeamento angular (trilha C) dos setores 2 a 5. No gráfico de coordenadas paralelas presente na trilha B é possível distinguir dois agrupamentos de dados: as coordenadas dos cones auxiliares em cada eixo, que são caracterizadas pelo valor mais alto da coordenada que coincide com o índice da trilha, e um agrupamento principal, que corresponde às coordenadas dos pontos na ROI.

A figura 52 exibe as soluções obtidas pelos algoritmos coneMOEA/D e coneNSGA-III em duas instâncias com dez objetivos utilizando o CAP-vis. As análises feitas na figura 51b se repetem na figura 52a pois a única mudança ocorre com o número de objetivos e com a escolha do ângulo, que passa a ser  $\theta_2$  e desta forma corresponde a uma ROI de menor extensão.

A figura 52b ilustra uma solução obtida pelo algoritmo coneNSGA-III no cenário onde a ROI é definida pelo cone obtido pelo vetor  $\mathbf{v}_1$  e ângulo  $\theta_3$ . O equilíbrio entre os objetivos e a extensão da ROI característicos desta escolha de ROI podem ser visualizados nesta figura.



(a) Solução obtida pelo algoritmo coneMOEA/D com cone definido pelo vetor  $\mathbf{v}_3$  e ângulo  $\theta_2$  (b) Solução obtida pelo algoritmo coneNSGAIII com cone definido pelo vetor  $\mathbf{v}_1$  e ângulo  $\theta_3$

Figura 52 – Soluções obtidas pelos algoritmos coneMOEA/D e coneNSGA-III no problema pd-MOP02 com 10 objetivos.

### 7.3 Considerações finais

Um dos desafios apresentados pela otimização com muitos objetivos é a seleção de um conjunto satisfatório de soluções obtidas no espaço dos objetivos. O excesso de informação presente na análise das soluções com elevado número de objetivos é tão prejudicial quanto a falta de informações. Um algoritmo evolutivo com uma população muito grande não apresenta um conjunto de soluções em um tempo razoável e a utilização de uma população de tamanho adequado produz um conjunto de soluções com uma baixa densidade de pontos no espaço dos objetivos quando este tem dimensão muito grande. De forma complementar, o tomador de decisão de maneira geral tem um bom conhecimento do problema e é capaz de estabelecer *a priori* características desejáveis e de pouco interesse que o conjunto de soluções apresentado pelo otimizador irá apresentar. A união destes dois fatores faz com que a escolha de uma região de interesse no espaço dos objetivos seja mais racional em problemas de otimização com muitos objetivos.

Os métodos que utilizam uma ASF apresentam bons resultados em problemas com dois e três objetivos mas exigem o conhecimento da localização da Fronteira Pareto do problema para a determinação ideal do ponto de referência utilizado. De maneira geral extensão e a localização da ROI nesta técnica é fortemente dependente de uma boa escolha deste ponto. No sentido oposto, a utilização do cone de vetores permite a definição da ROI de acordo com um conjunto de preferências do tomador de decisão, que pode atribuir um conjunto de pesos ao eixo do cone e definir a extensão da ROI por meio do ângulo de abertura  $\theta$ . Além disto o método proposto se mostrou eficiente nos problemas com muitos

---

objetivos analisados, tanto na pertinência das soluções na ROI quanto na convergência com a Fronteira Pareto definida pela ROI. Neste cenário, analisando o número de soluções obtidas na ROI, a comparação entre as abordagens *a priori* e *a posteriori* não deixa dúvida quanto a superioridade do método onde as preferências do tomador de decisão são incorporadas no processo de otimização.



## 8 Um Algoritmo coevolutivo para Otimização Robusta Multiobjetivo

Este capítulo apresenta um Algoritmo coevolutivo para Otimização Robusta Multiobjetivo por meio da minimização do pior caso que utiliza a estratégia da decomposição/agregação, chamado de *Coevolutionary Robust MOEA/D* (CR-MOEA/D) [99]. O algoritmo proposto utiliza como base o algoritmo MOEA/D. Sua escolha se justifica por ser um algoritmo eficiente, bem aceito pela comunidade científica e por apresentar de forma definida as etapas de decomposição e agregação, necessárias para o processo evolutivo proposto. Sua eficiência fez com que novos algoritmos baseados em decomposição/agregação fossem desenvolvidos e a metodologia coevolutiva competitiva apresentada neste trabalho pode ser aplicada em outros algoritmos desta classe.

O algoritmo proposto utiliza uma estratégia competitiva entre duas populações de soluções candidatas de mesmo tamanho: um conjunto  $X$  contendo a população de vetores  $\mathbf{x}$  do espaço de decisão e um conjunto  $\Delta$  contendo uma população de vetores de ruído  $\delta$  de pequena intensidade, com  $\delta \in [-\epsilon, \epsilon]$ . Cada vetor  $\mathbf{x}^i \in X$  está associado a um único vetor  $\delta^i \in \Delta$  determinando o elemento  $\mathbf{x}^i + \delta^i$ . A cada iteração do ciclo evolutivo duas etapas de otimização  $E_1$  e  $E_2$  são realizadas. Considerando o problema multiobjetivo de minimização da função  $F : \mathbb{R}^N \rightarrow \mathbb{R}^M$ , a primeira etapa busca soluções no espaço  $X$  que minimizam a função  $F(\mathbf{x} + \delta)$ , ou seja, a primeira etapa de cada processo evolutivo trata do problema de minimização

$$E_1 : \quad \mathbf{x} = \arg \min F(\mathbf{x} + \delta) \quad (8.1)$$

Nesta etapa o algoritmo faz a busca de valores ótimos da variável  $\mathbf{x} \in X$  procurando a minimização dos objetivos.

A segunda etapa busca soluções no espaço  $\Delta$  que maximizam a função  $F(\mathbf{x} + \delta)$ , ou seja, a segunda etapa de cada ciclo evolutivo trata do problema de maximização

$$E_2 : \quad \delta = \arg \max F(\mathbf{x} + \delta) \quad (8.2)$$

Nesta etapa o algoritmo faz a busca de valores ótimos da variável  $\delta \in \Delta$  procurando a maximização dos objetivos.

Uma vez que o ruído  $\delta$  é de pequena intensidade, o problema de maximização da segunda etapa procura um ponto na vizinhança de  $\mathbf{x}$  que maximiza a função  $F$  nesta região. Na iteração seguinte este ciclo se repete. Estas etapas estão continuamente em

conflito e sua competição conduz a um conjunto de soluções ótimas considerando o pior cenário, que corresponde ao valor de  $\delta$  que maximiza o valor da função  $F$  na vizinhança de  $\mathbf{x}$ .

Uma vez que o algoritmo proposto é uma variação do MOEA/D suas estruturas são similares e exigem poucas adaptações. O algoritmo pode ser dividido em duas etapas: Inicialização e Ciclo Evolucionário, que são descritos abaixo:

**Inicialização:** Nesta etapa duas populações de vetores são geradas de forma aleatória:

Uma população  $X = \{\mathbf{x}^1, \dots, \mathbf{x}^P\}$  com  $P$  vetores contida do espaço de decisão e uma população de vetores de ruído  $\Delta = \{\delta^1, \dots, \delta^P\}$  com  $P$  vetores de ruído. Os vetores de ruído  $\delta^i = (\delta_1^i, \dots, \delta_N^i)$  têm suas coordenadas  $\delta_j^i$  contidas no intervalo  $[-\epsilon_j, \epsilon_j]$ . O parâmetro  $\epsilon$  pode ser o mesmo para cada coordenada e indica a extensão da perturbação admitida em cada componente dos vetores  $\mathbf{x} \in X$  e é um parâmetro do problema. Também deve ser gerado um conjunto de vetores de peso  $W = \{\mathbf{w}^1, \dots, \mathbf{w}^P\}$ . Após a geração dos conjuntos  $X$  e  $\Delta$  é feita uma associação um-a-um entre seus elementos: cada vetor de  $\mathbf{x}^i \in X$  está associado a um único vetor de ruído  $\delta^i \in \Delta$ . Em seguida é gerada a estrutura de vizinhança  $B(i) = \{i_1 \dots i_T\}$ : cada vetor de peso  $\mathbf{w}^i \in W$  fica associado a seus  $T$  vizinhos mais próximos  $\mathbf{w}^{i_1}, \dots, \mathbf{w}^{i_T}$ . Esta associação pode ser feita utilizando a distância euclidiana entre os pontos que definem os vetores  $\mathbf{w}^i$  e  $\mathbf{w}^j$  ou o ângulo entre eles. O conjunto  $B(i)$  armazena o conjunto de índices dos vetores  $\mathbf{w}^{i_j}$  associados ao vetor  $\mathbf{w}^i$ . Em seguida avalie os elementos da população fazendo  $FV^i = F(\mathbf{x}^i + \delta^i)$ .

**Ciclo Evolucionário:** Em cada iteração do algoritmo duas instâncias de otimização E1 e E2 são realizadas.

**Instância E1:** Para cada vetor  $\mathbf{x}^i \in X$  dois vetores  $\mathbf{x}^{i_a}$  e  $\mathbf{x}^{i_b}$  são selecionados na vizinhança de  $\mathbf{x}^i$  definida pelo conjunto  $B(i)$ . Então um novo vetor  $\mathbf{x}^{i,c}$  é criado aplicando os operadores genéticos de recombinação e mutação. Em seguida é considerado o sub-problema de otimização escalar dado por

$$\min_{\mathbf{x}} g(\mathbf{x}|\delta^i, \mathbf{w}^i, \mathbf{z}) \quad (8.3)$$

Sendo  $g(\star)$  alguma função de agregação (Soma ponderada, norma de Tchebycheff ponderada, PBI etc). Em seguida, até que o limite de substituições seja alcançado, para cada elemento selecionado da população, se

$$g(\mathbf{x}^{i,c}|\delta^j, \mathbf{w}^j, \mathbf{z}) \leq g(\mathbf{x}^j|\delta^j, \mathbf{w}^j, \mathbf{z}) \quad (8.4)$$

faça  $\mathbf{x}^j = \mathbf{x}^{i,c}$  e  $FV^j = F(\mathbf{x}^{i,c} + \delta^j)$ .

**Instância E2:** Para cada vetor  $\delta^i \in \Delta$  dois vetores  $\delta^{i_a}$  e  $\delta^{i_b}$  são selecionados na vizinhança de  $\delta^i$  definida pelo conjunto  $B(i)$ . Então um novo vetor  $\delta^{i,c}$  é criado aplicando os operadores genéticos de recombinação e mutação. Em seguida é considerado o sub-problema de otimização escalar dado por

$$\max_{\delta} g(\delta|\mathbf{x}^i, \mathbf{w}^i, \mathbf{z}) \quad (8.5)$$

Sendo  $g(\star)$  alguma função de agregação (Soma ponderada, norma de Tchebycheff ponderada, PBI etc). Em seguida, até que o limite de substituições seja alcançado, para cada elemento selecionado da população, se

$$g(\delta^{i,c}|\mathbf{x}^j, \mathbf{w}^j, \mathbf{z}) \geq g(\delta^j|\mathbf{x}^j, \mathbf{w}^j, \mathbf{z}) \quad (8.6)$$

faça  $\delta^j = \delta^{i,c}$  e  $FV^j = F(\mathbf{x}^j + \delta^{i,c})$ .

As instâncias de otimização E1 e E2 são muito parecidas. A diferença essencial entre elas é que na instância E1 a otimização é feita pelo ajuste do vetor  $\mathbf{x} \in X$  no espaço de decisão enquanto que na instância E2 a otimização é feita pelo ajuste do vetor de ruído  $\delta \in \Delta$ . Além disso a relação de ordem nas equações 8.3 e 8.5 são invertidas pois em E1 está presente um sub-problema escalar de minimização enquanto que em E2 está presente um sub-problema escalar de maximização. A estratégia de decomposição presente no MOEA/D faz com o que o processo co-evolutivo seja simples de se implementar e entender uma vez que as instâncias de otimização E1 e E2 são independentes e percorrem todos os elementos dos conjuntos de variáveis  $X$  e  $\Delta$ .

A figura 53 ilustra a ação das duas instâncias de otimização E1 e E2 em um vetor  $A = F(\mathbf{x}^0 + \delta^0)$  no espaço dos objetivos de um Problema de Otimização Multiobjetivo (POM) com dois objetivos. Após a primeira instância de otimização E1 uma nova solução  $B = F(\mathbf{x}^1 + \delta^0)$  foi obtida com a obtenção do novo vetor  $\mathbf{x}^1$  que minimiza os objetivos 1 e 2. Em seguida esta nova solução  $B$  é submetida à segunda instância de otimização E2 e uma terceira solução  $C = F(\mathbf{x}^1 + \delta^1)$  é obtida com a definição do vetor  $\delta^1$  que maximiza os objetivos em uma vizinhança de  $\mathbf{x}^1$ , considerando o pior caso para o problema  $F$  nesta região. A linha tracejada corresponde à Fronteira Pareto do problema e a linha contínua corresponde à Fronteira Pareto Robusta.

Um resumo das etapas do algoritmo  $C_RMOEA/D$  é apresentado abaixo:

#### A) Inicialização:

- 1) Gere a população inicial de  $P$  vetores  $X = \{\mathbf{x}^1, \dots, \mathbf{x}^P\}$  de forma aleatória no espaço de decisão  $\mathbb{R}^N$ ;

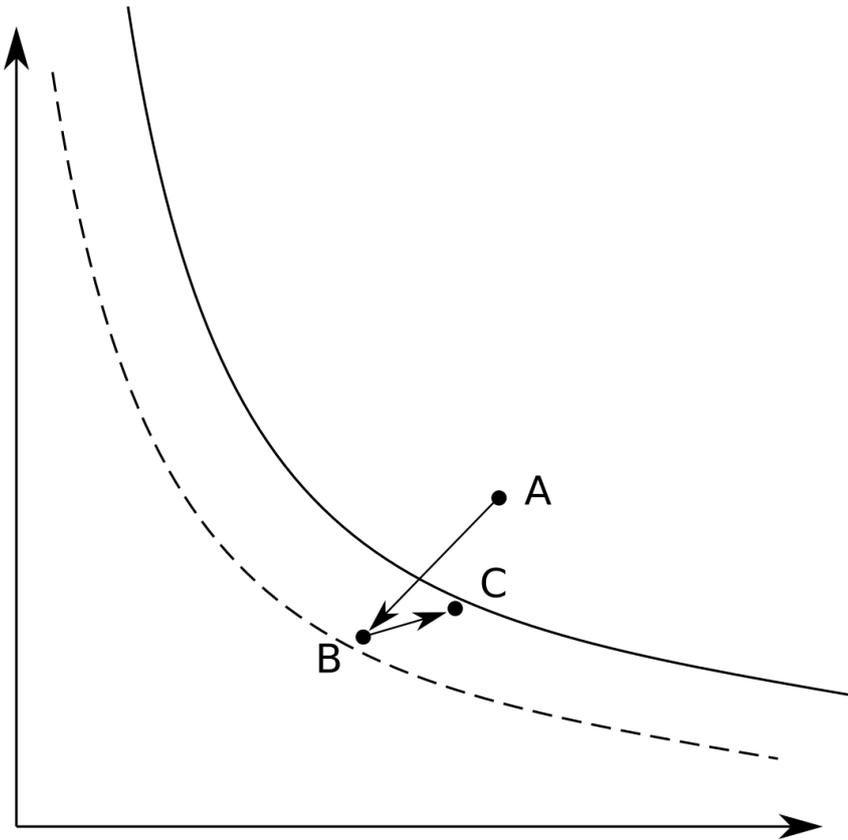


Figura 53 – Ação das duas instâncias de otimização E1 e E2 em um vetor  $A$  no espaço dos objetivos.

- 2) Gere a população inicial de  $P$  vetores de ruído  $\Delta = \{\delta^1, \dots, \delta^P\}$  de forma aleatória no espaço de decisão  $\mathbb{R}^N$ ;
- 3) Associe cada vetor  $\mathbf{x}^i \in X$  a um único vetor  $\delta^i \in \Delta$ ;
- 4) Gere  $P$  vetores de peso  $W = \{\mathbf{w}^1, \dots, \mathbf{w}^P\}$  no espaço dos objetivos  $\mathbb{R}^M$ ;
- 5) Associe cada vetor  $\mathbf{x}^i \in X$  a um único vetor  $\mathbf{w}^i \in W$ ;
- 6) Para cada vetor  $\mathbf{w}^i \in W$  gere a estrutura de vizinhança  $B(i) = \{i_1, \dots, i_T\}$  considerando os  $T$  vetores  $\mathbf{w}^j \in W$  mais próximos de  $\mathbf{w}^i$  utilizando a distância euclidiana entre os pontos que definem os vetores  $\mathbf{w}^i$  e  $\mathbf{w}^j$  ou o ângulo entre eles.
- 7) Avalie os elementos da população fazendo  $FV^i = F(\mathbf{x}^i + \delta^i)$ , com  $1 \leq i \leq P$ .
- 8) Determine a estimativa da solução utópica  $\mathbf{z}$ .

B) **Ciclo Evolucionário:** Enquanto o critério de parada não for atingido, execute sequencialmente as instâncias de otimização E1 e E2:

**E1:** Para cada vetor  $\mathbf{x}^i \in X$  faça:

- 1) Selecione dois indivíduos  $\mathbf{x}^{i_a}$  e  $\mathbf{x}^{i_b}$  com  $\{i_a, i_b\} \subset B(i)$  e gere um novo indivíduo  $\mathbf{x}^{i,c}$  utilizando os operadores genéticos de recombinação e mutação;

- 2) Até que o limite de substituições seja alcançado, se  $g(\mathbf{x}^{i,c}|\delta^j, \mathbf{w}^j, \mathbf{z}) \leq g(\mathbf{x}^j|\delta^j, \mathbf{w}^j, \mathbf{z})$  faça  $\mathbf{x}^j = \mathbf{x}^{i,c}$  e  $FV^j = F(\mathbf{x}^{i,c} + \delta^j)$ .

**E1:** Para cada vetor  $\delta^i \in \Delta$  faça:

- 1) Selecione dois indivíduos  $\delta^{i_a}$  e  $\delta^{i_b}$  com  $\{i_a, i_b\} \subset B(i)$  e gere um novo indivíduo  $\delta^{i,c}$  utilizando os operadores genéticos de recombinação e mutação;
- 2) Até que o limite de substituições seja alcançado, se  $g(\delta^{i,c}|\mathbf{x}^j, \mathbf{w}^j, \mathbf{z}) \geq g(\delta^j|\mathbf{x}^j, \mathbf{w}^j, \mathbf{z})$  faça  $\delta^j = \delta^{i,c}$  e  $FV^j = F(\mathbf{x}^j + \delta^{i,c})$ .

## 8.1 Resultados

Esta seção apresenta o resultado de uma série de experimentos realizados que procuram verificar a eficiência do algoritmo coevolutivo proposto em encontrar soluções robustas para problemas de otimização multiobjetivo e com muitos objetivos, de forma usual ou em uma região específica do espaço dos objetivos (ROI). Para isto foram contrapostos os resultados obtidos pelo algoritmo proposto e pelo algoritmo NSGA-III usual, aplicados na versão robusta do problema poMOP02. Mais uma vez a escolha deste problema se justifica por sua adequação a problemas com qualquer número de objetivos. Foram realizadas 30 instâncias de cada cenário.

Nos problemas onde se buscam soluções em todo o espaço dos objetivos os vetores de peso foram gerados pelo método de Scheffé. Nos problemas que procuram explorar uma ROI os vetores de peso foram gerados pelo método proposto no capítulo 6. Uma vez que a eficácia da aplicação de cone de vetores de preferência na busca de soluções em uma ROI já foi verificada no capítulo 7, as regiões de interesse analisadas nesta seção se restringem às regiões apresentadas na tabela 8.1, sendo  $\Theta$  o ângulo entre a hiper-diagonal  $\mathbf{d} = (1, \dots, 1)$  e os eixos da base canônica  $e_i$  do espaço dos objetivos  $\mathbb{R}^M$ . Tanto nos problemas de otimização de caráter geral quanto nos problemas que visam à exploração de uma ROI serão considerados os casos com 3, 5 e 10 objetivos.

Em todos os cenários foi utilizada uma população de 300 indivíduos e o número de avaliações das funções objetivo foi limitado a  $1e5$ . Cada ponto no espaço de decisão tem a forma  $\mathbf{x} = (\mathbf{x}^p, \mathbf{x}^d)$ , sendo  $\mathbf{x}^p$  um vetor com  $M - 1$  posições responsável pela posição relativa dos pontos no espaço dos objetivos e  $\mathbf{x}^d$  um vetor com 10 posições responsável

ROI	Vetor	Ângulo de abertura do cone
ROI 1	$\mathbf{v}_1 = (1, \dots, 1)$	$\theta_1 = 0.05 \times \Theta$
ROI 2	$\mathbf{v}_2 = (3, 1, \dots, 1)$	$\theta_2 = 0.15 \times \Theta$
ROI 3	$\mathbf{v}_3 = (1, 3, \dots, 3)$	$\theta_3 = 0.25 \times \Theta$

Tabela 12 – Regiões de Interesse utilizadas nos testes realizados

Parâmetro	Valor
População	300
Iterações	1e5
Penalidade $\theta$ da função de agregação (PBI)	5
Fator de dispersão da recombinação ( $\mu_c$ )	25
Fator de dispersão da mutação ( $\mu_m$ )	20
Raio $\epsilon$ do intervalo de perturbação	0.05
Número de objetivos (M)	3,5,10
Número de variáveis de decisão	$M + 9$

Tabela 13 – Valores dos parâmetros utilizados nos experimentos

pela convergência da população na Fronteira Pareto. Foi utilizada a função de agregação PBI com penalidade  $\theta = 5$ . No algoritmo CR-MOEA/D o vetor de perturbação  $\delta$  tem a forma  $\delta = (\delta^p, \delta^d)$ , sendo  $\delta^p = (0, \dots, 0)$  e  $\delta^d = (\delta_1^d, \dots, \delta_{10}^d)$  com  $\delta_i^d$  uma variável aleatória com distribuição uniforme no intervalo  $(-0.05, 0.05)$ . Estas informações são apresentadas na tabela 13.

Na avaliação dos resultados foram utilizadas as métricas IGD e RDM. Nas instâncias que fazem a exploração de uma ROI uma amostra desta região foi gerada utilizando um cone de vetores como descrito no capítulo 7 para uso na métrica IGD. O problema pdMOP02 robusto apresenta duas Fronteiras Pareto: uma usual (PF), que trata do otimização geral sem a preocupação da robustez das soluções, e uma Fronteira Pareto Robusta (Robust Pareto Front (RPF)), onde as soluções apresentam pouca variação na presença de ruído nas variáveis do espaço de decisão. Estes conjuntos serão denotados por PF e RPF respectivamente. A PF deste problema é obtida fazendo  $\mathbf{x}^d = (0.6001, \dots, 0.6001)$  nas variáveis de decisão  $\mathbf{x} = (\mathbf{x}^p, \mathbf{x}^d)$  enquanto que na RPF as coordenadas do vetor  $\mathbf{x}^d = (x_1^d, \dots, x_{10}^d)$  podem variar no intervalo  $(0.05, 0.35)$ . Na RPF utilizada nos testes foi utilizado  $x_i^d = 0.02$  com  $1 \leq i \leq 10$ . O algoritmo CR-MOEA/D foi avaliado segundo sua convergência na RPF enquanto que o algoritmo NSGA-III foi avaliado segundo sua convergência na PF.

A robustez das soluções foi avaliada da seguinte maneira:

1. Foram coletadas as soluções  $\mathbf{x} = (\mathbf{x}^p, \mathbf{x}^d)$  no espaço de decisão de cada instância. No algoritmo CR-MOEA/D foram coletadas apenas os valores da população principal, sendo desprezados os valores da população secundária.;
2. Para cada elemento da população  $P$  foi gerado um vetor de ruído  $\delta = (\delta_1, \dots, \delta_{10})$  com  $\delta_i$  uma variável aleatória com distribuição uniforme no intervalo  $(-0.05, 0.05)$ . Em seguida é gerada uma nova população  $P'$  formada pelos vetores  $\mathbf{x}' = (\mathbf{x}_p, \mathbf{x}_d + \delta)$
3. A nova população  $P'$  é avaliada e os resultados obtidos são submetidos às métricas

CR-MOEA/D				
Objetivos	IGD		RDM	
	Solução Obtida	Solução com ruído	Solução Obtida	Solução com ruído
3	5.110e-01	5.108e-01	2.427e-02	1.823e-02
5	6.404e-01	6.401e-01	2.567e-02	1.908e-02
10	8.710e-01	8.702e-01	2.953e-02	2.080e-02

NSGA-III				
Objetivos	IGD		RDM	
	Solução Obtida	Solução com ruído	Solução Obtida	Solução com ruído
3	4.980e-01	1.091	2.048e-02	1.287
5	5.542e-01	1.257	2.064e-02	1.438
10	6.601e-01	1.381	2.814e-02	1.560

Tabela 14 – Métricas IGD e RDM para o problema robusto poMOP02 com e sem a presença de ruído na variável  $\mathbf{x}^d$

IGD e RDM. As soluções obtidas pelo algoritmo CR-MOEA/D utilizam a RPF como referência enquanto que as soluções obtidas pelo algoritmo NSGA-III utilizam a PF como referência na métrica IGD.

A tabela 14 apresenta os valores das métricas obtidas pelos algoritmos CR-MOEA/D e NSGA-III nas instâncias analisadas para os problemas sem ROI. Em todas as situações as métricas IGD e RDM apresentam uma degradação natural com o aumento do número de objetivos mas ambos algoritmos apresentam uma boa convergência e diversidade nas respectivas Fronteiras Pareto. As soluções obtidas sem a presença do ruído o algoritmo NSGA-III apresentam uma melhor performance em geral. Entretanto, a presença de ruído provoca uma nítida degradação na qualidade das soluções obtidas pelo algoritmo NSGA-III, caracterizando desta maneira suas soluções como soluções não robustas. Já a presença do ruído nas soluções obtidas pelo algoritmo CR-MOEA/D provocou um efeito inverso: houve uma melhora da convergência das soluções obtidas em todos os cenários. Este fenômeno era esperado uma vez que no algoritmo coevolucionário a segunda instância de otimização procura maximizar o valor das funções por meio do valor ótimo da variável de perturbação. Como o ruído adicionado é aleatório esta otimalidade foi perdida e houve uma redução do valor nominal dos objetivos. De qualquer forma o valor obtido sem a adição de ruído e com o valor ótimo da perturbação obtido corresponde ao pior caso (*worst case*) para este problema no intervalo considerado, que é a abordagem proposta pelo algoritmo CR-MOEA/D.

O resultado dos experimentos que buscam soluções em uma ROI são apresentados nas tabelas 15 e 16. A tabela 15 apresenta o tamanho da população encontrada na ROI e

Objetivos	# ROI	# R1	# R2
<b>ROI 1</b>			
3	282.4	17.6	0.0
5	280.2	17.4	0.3
10	209.3	28.2	9.1
<b>ROI 2</b>			
3	281.5	18.5	0.0
5	278.7	19.0	0.3
10	180.1	43.7	17.6
<b>ROI 3</b>			
3	279.6	18.0	0.3
5	279.2	20.8	0.0
10	151.9	71.6	17.9

Tabela 15 – Tamanho da população encontrada na ROI e nas vizinhanças R1 e R2 obtida pelos algoritmos CR-MOEA/D no problema robusto pdMOP02

nas mesmas vizinhanças R1 e R2 descritas no capítulo 7 pelo algoritmo CR-MOEA/D. Em todas as situações as soluções se concentram na ROI, com um forte decréscimo nas regiões R1 e R2. Estes são análogos aos experimentos realizados no capítulo 7 e indicam que a adição de uma instância de otimização no algoritmo coevolucionário não afetou a convergência da população na região de interesse ou em suas vizinhanças em nenhum dos casos analisados.

A tabela 16 apresenta o valor as métricas IGD e RDM para as soluções obtidas pelo algoritmo CR-MOEA/D no problema robusto poMOP02 com 3, 5 e 10 objetivos. Em cada métrica foi considerada a Fronteira Pareto Robusta (FPR) e foram comparados os resultados com e sem a adição de ruído nas soluções obtidas. Em todos os cenários é observada a progressão dos valores das métricas com o aumento do número de objetivos, o que mais uma vez é um resultado esperado uma vez que a complexidade do problema é proporcional ao número de objetivos. Também se observa a conformidade dos valores das métricas IGD e RDM nas soluções sem a adição de ruído, o que indica que a métrica RDM não apresenta valores contraditórios com a métrica IGD nos problemas analisados. Entretanto, a métrica proposta apresenta a vantagem de não depender de uma particular amostra da Fronteira Pareto do problema. Os resultados apresentados nesta tabela indicam que as soluções obtidas pelo algoritmo CR-MOEA/D preservam a convergência com a presença do ruído nas variáveis de decisão. Mais uma vez foi observado o fenômeno da pequena redução dos valores das métricas nesta situação e discutida anteriormente, que caracteriza as soluções obtidas por este algoritmo como de fato prevendo o pior caso em cada cenário.

As figuras 54 a 61 ilustram os resultados obtidos em algumas instâncias utilizando

Objetivos	IGD		RDM	
	Solução Obtida	Solução com ruído	Solução Obtida	Solução com ruído
<b>ROI 1</b>				
3	2.669e-01	2.660e-01	1.344e-02	1.048e-02
5	2.229e-01	2.218e-01	1.467e-02	1.173e-02
10	3.943e-01	3.926e-01	2.168e-02	1.602e-02
<b>ROI 2</b>				
3	1.809e-01	1.800e-01	1.297e-02	1.026e-02
5	2.753e-01	2.744e-01	1.638e-02	1.266e-02
10	5.916e-01	5.901e-01	2.408e-02	1.790e-02
<b>ROI 3</b>				
3	1.325e-01	1.317e-01	1.245e-02	1.009e-02
5	3.228e-01	3.221e-01	1.828e-02	1.418e-02
10	8.083e-01	8.064e-01	3.234e-02	2.307e-02

Tabela 16 – Métricas IGD e RDM para as soluções obtidas pelo algoritmo CR-MOEA/D no problema robusto pdMOP02 com e sem a presença de ruído na variável  $\mathbf{x}^d$

o CAP-vis. Em cada figura a solução obtida é representada utilizando elementos azuis (pontos e curvas) enquanto que as respectivas Fronteiras Pareto (usual e robusta) são representadas em todos os setores apenas na trilha C por meio do mapeamento angular utilizando pontos pretos. A restrição na representação das Fronteiras Pareto apenas na trilha C se justifica pela melhor visualização das soluções nas outras trilhas. O objetivo da representação conjunta das soluções obtidas e das Fronteiras Pareto na trilha C é observar a posição relativa destes elementos, comparando a convergência das soluções na ROI e sua convergência.

A figura 54 apresenta as soluções obtidas pelo algoritmo NSGA-III no problema robusto pdMOP02 com e sem a presença do ruído na variável  $\mathbf{x}_d$ . A figura 54a ilustra as soluções obtidas sem a presença do ruído, que estão bem distribuídas ao longo de toda a Fronteira Pareto e apresentam uma boa convergência. De fato, a proximidade das soluções com a Fronteira Pareto é tal que suas representações se confundem. Esta ausência de clareza desaparece na figura 54b que representam as soluções obtidas com a adição de ruído nas variáveis  $\mathbf{x}_d$ . Esta diferença fica mais clara ao se analisar a escala vertical do mapeamento angular, que representa a norma dos vetores no espaço de soluções. Enquanto que na figura 54a esta escala varia de 1 a 1.0001, na figura 54b esta escala varia de 1 a 7, o que indica a completa deterioração da convergência das soluções obtidas pelo algoritmo NSGA-III na presença de incertezas no valor de suas soluções ótimas.

A figura 55 apresenta as soluções obtidas pelo algoritmo CR-MOEA/D no problema robusto pdMOP02 com e sem a presença do ruído na variável  $\mathbf{x}_d$ . Assim como no caso anterior, as soluções obtidas apresentam uma boa distribuição na Fronteira Pareto Robusta.

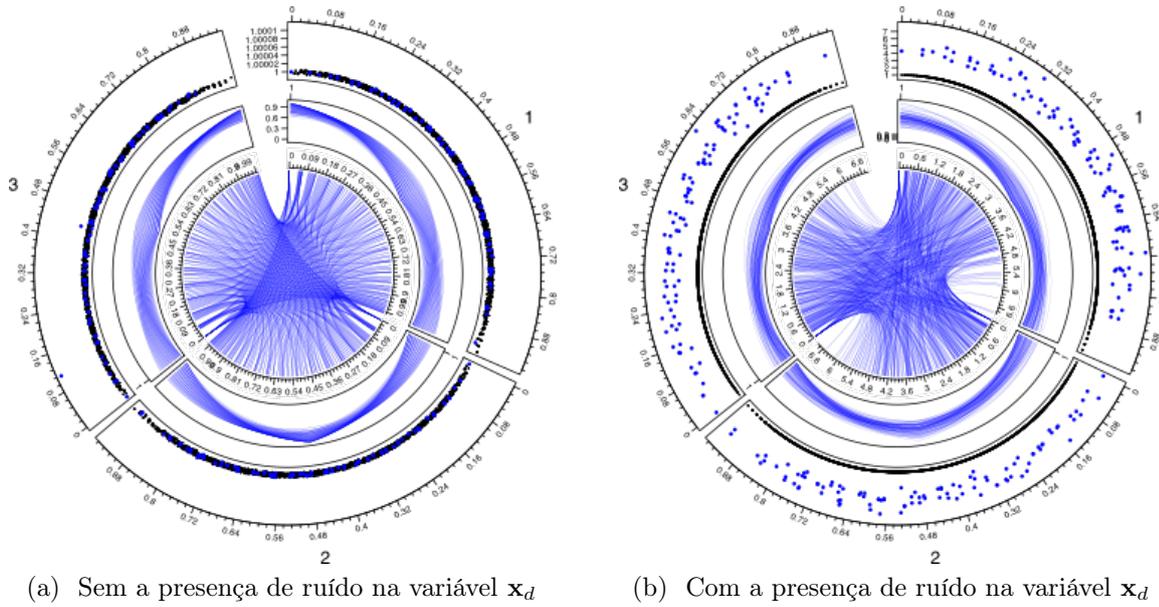


Figura 54 – Soluções obtidas pelo algoritmo NSGA-III no problema robusto pdMOP02.

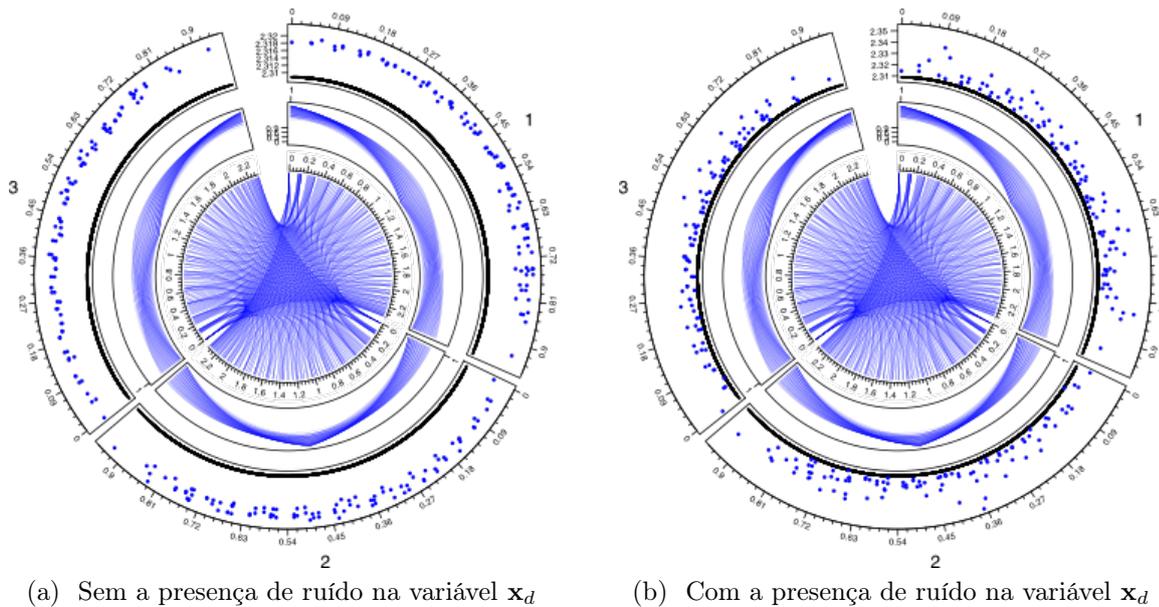


Figura 55 – Soluções obtidas pelo algoritmo CR-MOEA/D no problema robusto pdMOP02.

A figura 55a apresenta as soluções obtidas com o valor ótimo do ruído enquanto que a figura 55b apresenta as mesmas soluções com um ruído aleatório. Apesar da clara distinção entre os pontos da Fronteira Pareto Robusta e as soluções obtidas, a escala vertical do mapeamento angular mostra que estes conjuntos estão próximos e que de fato a adição de ruído aleatório não provocou o deterioramento da convergência das soluções.

As figuras 56 e 57 apresentam as soluções obtidas pelos algoritmos coneNSGA-III e coneCR-MOEA/D no problema robusto pdMOP02 com três objetivos na ROI R1 definida pelo cone de vetores com eixo  $\mathbf{v} = (1, 1, 1)$  e ângulo de abertura  $\theta = 0.05\Theta$  onde  $\Theta = \arccos\left(\frac{1}{\sqrt{3}}\right)$ .

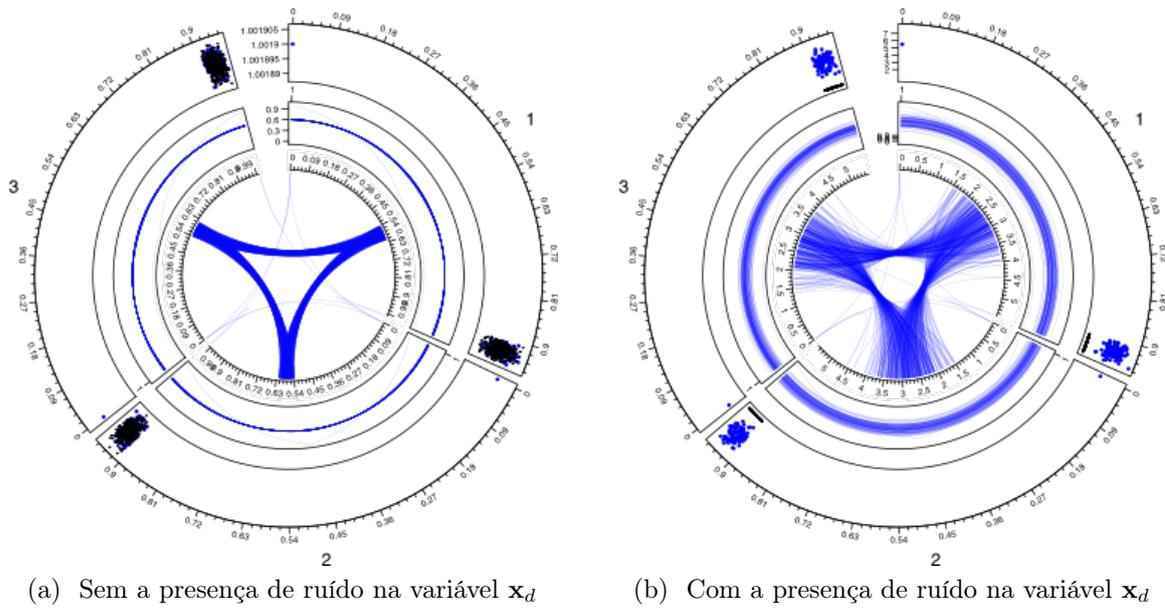


Figura 56 – Soluções obtidas pelo algoritmo NSGA-III na ROI R1 do problema robusto pdMOP02.

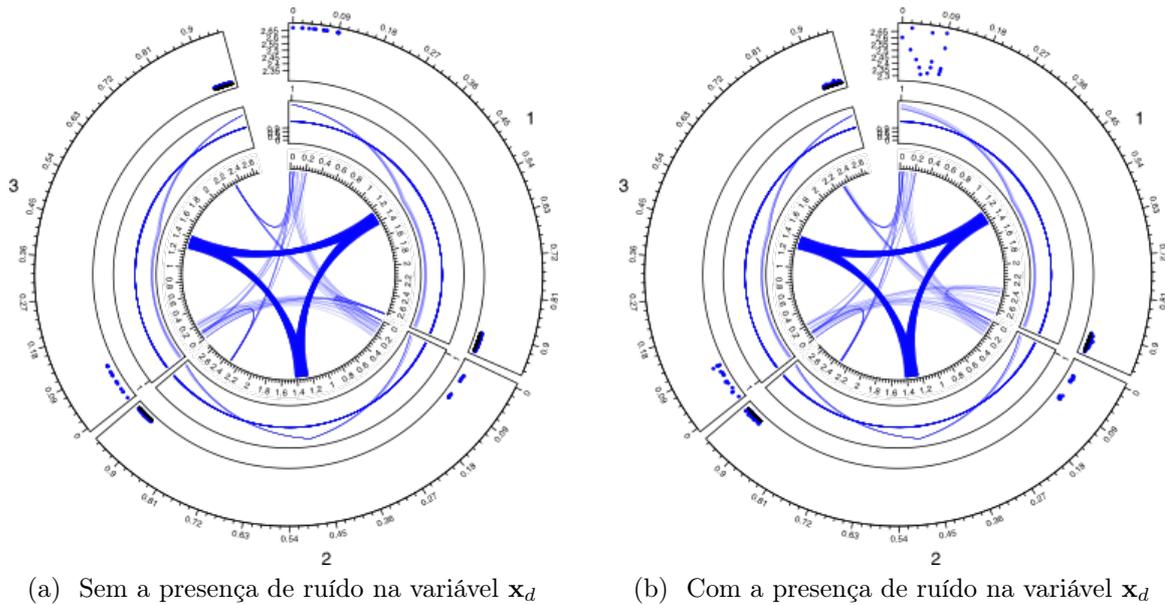


Figura 57 – Soluções obtidas pelo algoritmo CR-MOEA/D na ROI R1 do problema robusto pdMOP02.

A figura 56a apresenta as soluções obtidas pelo algoritmo coneNSGA-III nesta ROI. Nesta figura podemos observar o alinhamento das soluções obtidas com a ROI da Fronteira Pareto. O diagrama de cordas revela a reduzida faixa de variação das coordenadas de cada objetivo, ilustrado o quão restritiva é esta ROI. No mapeamento angular os pontos da Fronteira Pareto se misturam com os pontos as soluções obtidas, indicando a elevada taxa de convergência obtida. Apesar da representação gráfica do mapeamento angular apresentar uma espessura notável do bloco de pontos localizados na ROI, sua variação

é muito pequena: os extremos da escala do eixo vertical registram os valores 1.00189 e 1.001906, o que representa uma variação de  $1.6e-5$ . Todos os vetores da Fronteira Pareto tem norma igual a 1. Esta flutuação no valor da norma dos vetores pode ser explicada pela propagação de erros de arredondamento na normalização dos vetores da Fronteira Pareto e pela instabilidade numérica na expressão que calcula a norma destes vetores nos *softwares* utilizados <sup>1</sup>.

A figura 56b apresenta as soluções obtidas nesta instância do algoritmo coneNSGA-III com a presença de ruído nas variáveis  $\mathbf{x}_d$ . nesta figura a variação nos valores das normas da Fronteira Pareto não é perceptível e observa-se com clareza a deterioração da convergência da população, que pode ser medida pela escala vertical do mapeamento angular cujos valores variam de 1 a 7.

A figura 57 apresenta as soluções obtidas pelo algoritmo coneNSGA-III nesta ROI. Assim como nas soluções obtidas pelo algoritmo coneNSGA-III, as soluções obtidas se encontram alinhadas com a ROI. A presença do ruído não afetou a convergência das soluções como se pode observar comparando as figuras 57a e 57b. Também estão presentes nestas figuras os cones de vetores auxiliares. No setor 1 da figura 57a podemos observar que os pontos associados ao cone de vetores auxiliar deste eixo não obtiveram uma convergência satisfatória. Em consequência disto, estes mesmo pontos, quando adicionado o ruído, apresentaram uma grande degradação de sua convergência como pode ser observado na figura 57b. Entretanto, este fato não interferiu na robustez das soluções localizadas na ROI, que apresentaram uma taxa de convergência similar às soluções sem a presença de ruído.

As figuras 58 e 59 apresentam as soluções obtidas pelos algoritmos coneNSGA-III e coneCR-MOEA/D no problema robusto pdMOP02 com 5 objetivos na ROI R2 definida pelo cone de vetores com eixo  $\mathbf{v} = (3, 1, 1, 1, 1)$  e ângulo de abertura  $\theta = 0.15\Theta$  onde  $\Theta = \arccos\left(\frac{1}{\sqrt{5}}\right)$ .

As figuras 58a e 58b apresentam as soluções do algoritmo coneNSGA-III com e sem a presença de ruído na variável  $\mathbf{x}_d$  nesta ROI. Assim como nas situações anteriores a população convergiu corretamente na ROI e apresentou uma boa convergência no caso em que o ruído não está presente. Mais uma vez a presença de ruído provocou a degradação da convergência das soluções. Um efeito desta degradação pode ser observada comparando o diagrama de cordas presente nestas figuras. Na figura 58a as variáveis associadas ao setor 1 apresentam uma pequena variação, que é característico da ROI definida. Esta mesma porção do gráfico na figura 58b apresenta uma variação muito maior, o que mostra que o impacto da não robustez das soluções foi mais intenso no primeiro objetivo. Os demais objetivos apresentaram uma variação menos significativa na extensão da variação dos seus valores.

<sup>1</sup> A otimização foi feita no MATLAB e o CAP-vis é implementado no R.

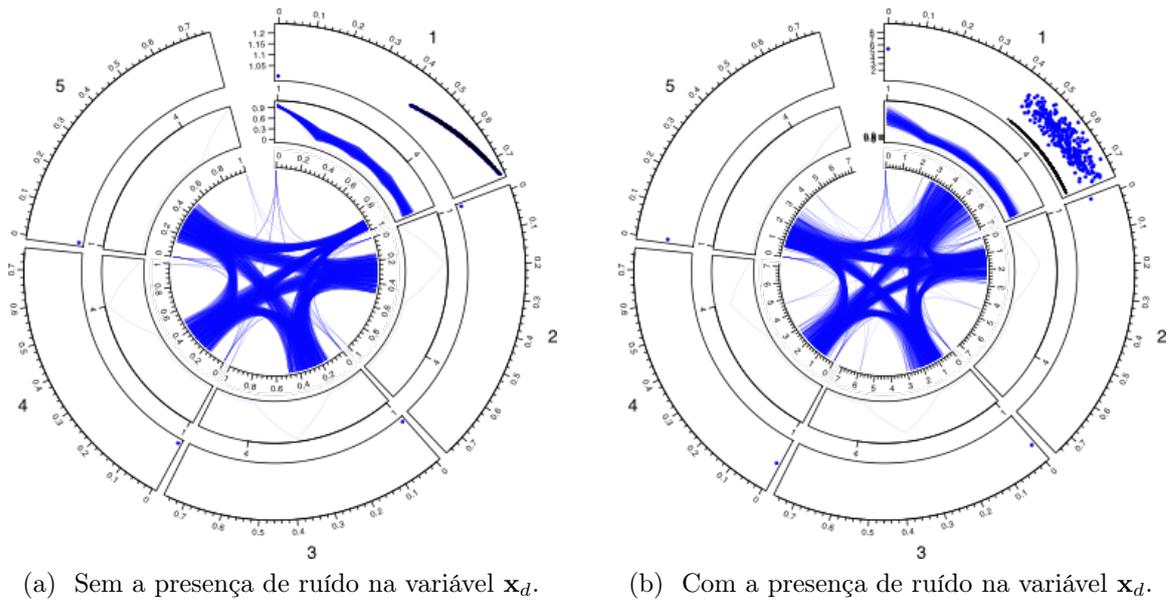


Figura 58 – Soluções obtidas pelo algoritmo NSGA-III na ROI R2 do problema robusto pdMOP02.

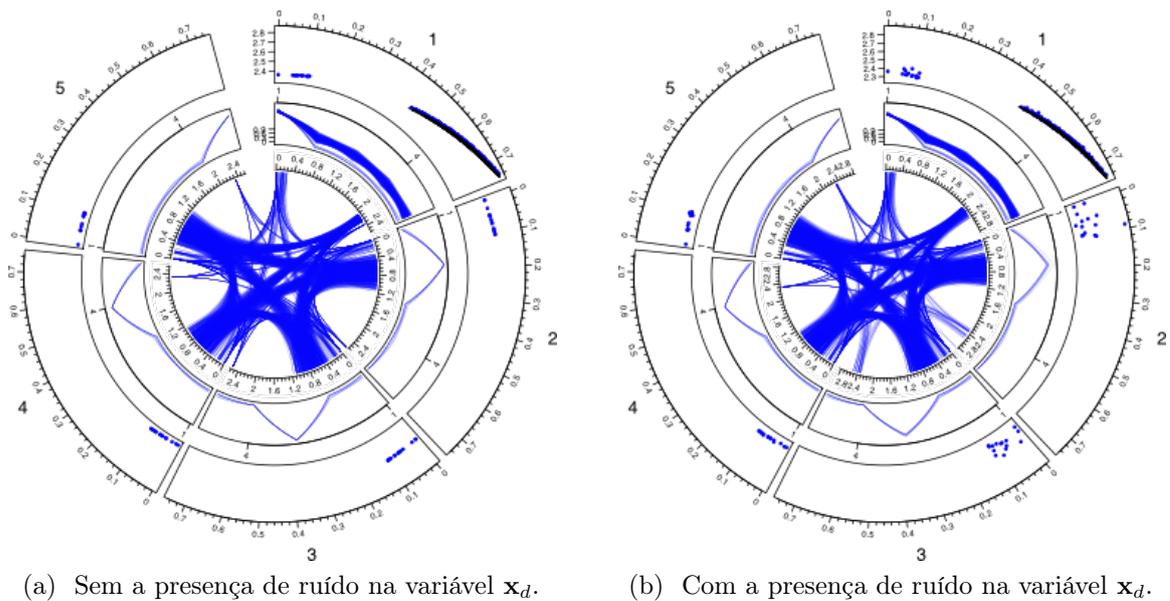


Figura 59 – Soluções obtidas pelo algoritmo CR-MOEA/D na ROI R2 do problema robusto pdMOP02.

As figuras 59a e 59b apresentam as soluções do algoritmo coneCR-MOEA/D com e sem a presença de ruído na variável  $x_d$  nesta ROI. Da mesma forma que os casos anteriores é perceptível a convergência da população na ROI e o pouco impacto causado pela presença do ruído nas variáveis  $x_d$ . Mais uma vez os elementos da população associados aos cones de vetores presentes nos eixos  $e_2$  e  $e_3$  não convergiram para a solução robusta, mas este fenômeno não afetou a convergência da população associada ao cone principal de vetores, tanto no quesito de pertinência das soluções na ROI quanto na convergência na mesma.

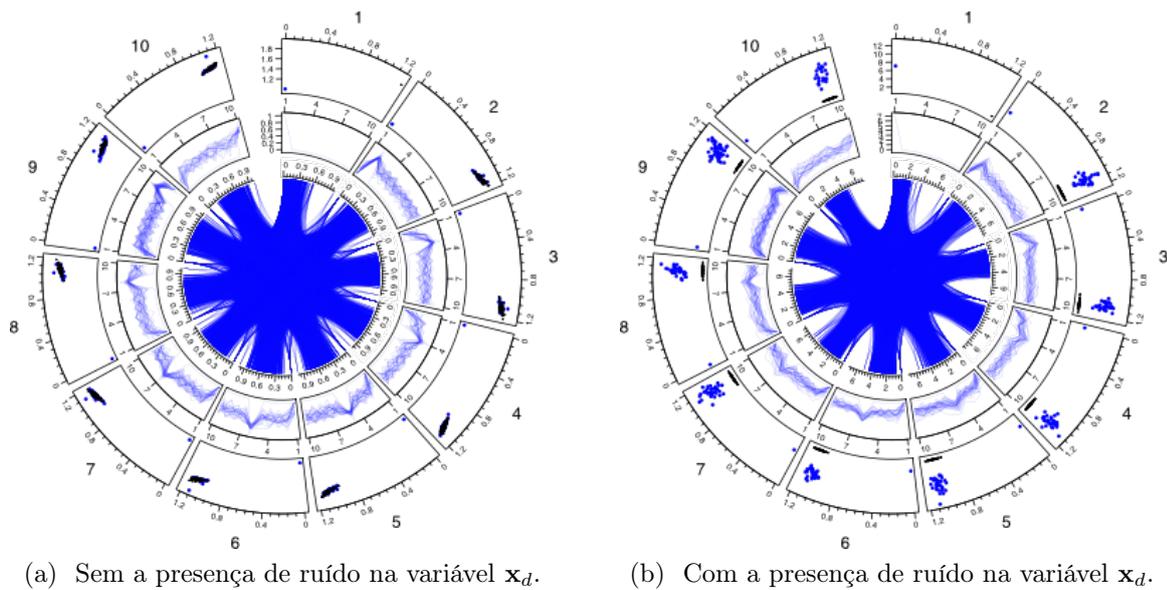


Figura 60 – Soluções obtidas pelo algoritmo NSGA-III na ROI R3 do problema robusto pdMOP02.

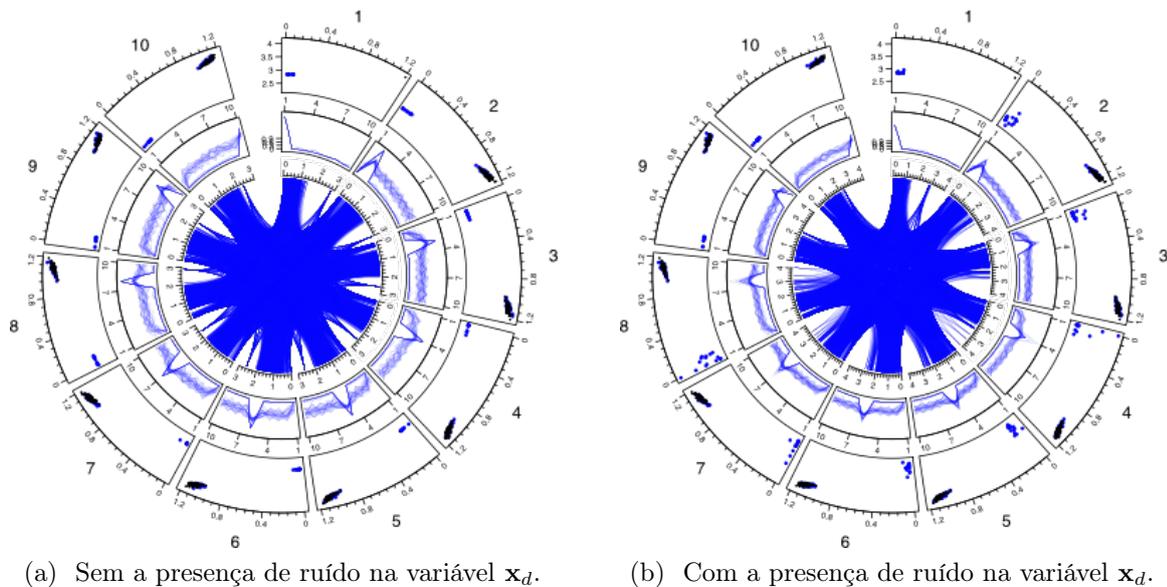


Figura 61 – Soluções obtidas pelo algoritmo CR-MOEA/D na ROI R3 do problema robusto pdMOP02.

As figuras 60 e 61 apresentam as soluções obtidas pelos algoritmos coneNSGA-III e coneCR-MOEA/D no problema robusto pdMOP02 com 10 objetivos na ROI R3 definida pelo cone de vetores com eixo  $\mathbf{v} = (1, 3, \dots, 3)$  e ângulo de abertura  $\theta = 0.25\Theta$  onde  $\Theta = \arccos\left(\frac{1}{\sqrt{10}}\right)$ .

Os resultados obtidos reparam as situações ilustradas anteriormente: em ambos os algoritmos a população apresenta uma boa pertinência na ROI e sem a presença do ruído os mesmos apresentam uma boa convergência e a adição do ruído na variável  $x_d$  afeta a convergência da solução obtida pelo algoritmo coneNSGA-III como pode ser observado

Parâmetro	Descrição	Intervalo de variação
$P_1$	Profundidade da ranhura do enrolamento (do estator) (mm)	$0.5 \leq P_1 \leq 18$
$P_2$	Comprimento da ranhura do enrolamento (mm)	$4 \leq P_2 \leq 21.5$
$P_3$	Ângulo inferior da ranhura do enrolamento (deg)	$1 \leq P_3 \leq 14$
$P_4$	Ângulo superior da ranhura do enrolamento (deg)	$1 \leq P_4 \leq 14$
$P_5$	Profundidade (da barra) de ímã permanente (mm)	$6 \leq P_5 \leq 26$
$P_6$	Espessura (da barra) de ímã permanente (mm)	$0.5 \leq P_6 \leq 20$
$P_7$	Largura (da barra) de ímã permanente (mm)	$0.5 \leq P_7 \leq 39$
$P_8$	Largura inferior da ranhura do rotor (mm)	$0.5 \leq P_8 \leq 39$
$P_9$	Largura superior da ranhura do rotor (mm)	$0.5 \leq P_8 \leq 39$

Tabela 17 – Parâmetros de simulação do motor de ímã permanente

nas figuras 60a e 60b, mas não afeta as soluções obtidas pelo algoritmo coneCR-MOEA/D como pode ser observado nas figuras 61a e 61b.

## 8.2 Otimização Robusta do motor IPM

Esta seção apresenta a aplicação do algoritmo CD-MOEA/D no problema de otimização robusta de um motor de ímã permanente (*Interior Permanent Magnet (IPM)*) [100], utilizando uma simulação do projeto de um motor, fornecida pelo *software* de simulação MagNet [62]. O motor contém um estator e um rotor confeccionados em aço, com densidade de corrente fixa a  $0.1528A/mm^2$  a  $200 Hz$ . Uma seção transversal deste motor é ilustrada na figura 62, que apresenta seus principais elementos, que são descritos na tabela 17. A relação entre os elementos utilizados no motor é discutida em [133]

Uma vez definido um conjunto de parâmetros  $P_1, \dots, P_9$ , a obtenção das características funcionais deste motor por meio do *software* de simulação tem um custo computacional elevando. A fim de reduzir o tempo gasto em cada simulação e fazer da etapa de otimização em um tempo razoável, utilizou-se uma rede neural de regressão generalizada (GRNN [135]) que faz um ajuste da superfície de resposta dada pelo simulador. Esta rede foi treinada com 21.523 amostras distintas do espaço de decisão. A escolha deste método de regressão foi motivada pela reduzida taxa de erro.

De posse do modelo matemático dado pela rede neural, foi realizada a etapa de otimização deste modelo. Utilizou-se o algoritmo proposto, com as mesmas configurações apresentadas na tabela 1, exceto o número de variáveis de decisão e o raio  $\epsilon$  que define a perturbação  $\Delta$  nas variáveis de decisão do problema. Nas simulações realizadas, foi considerado um intervalo de 4% do comprimento do intervalo de cada variável como sendo o comprimento da variável de perturbação associada a cada variável.

As nove variáveis do problema de otimização robusta do projeto deste motor são

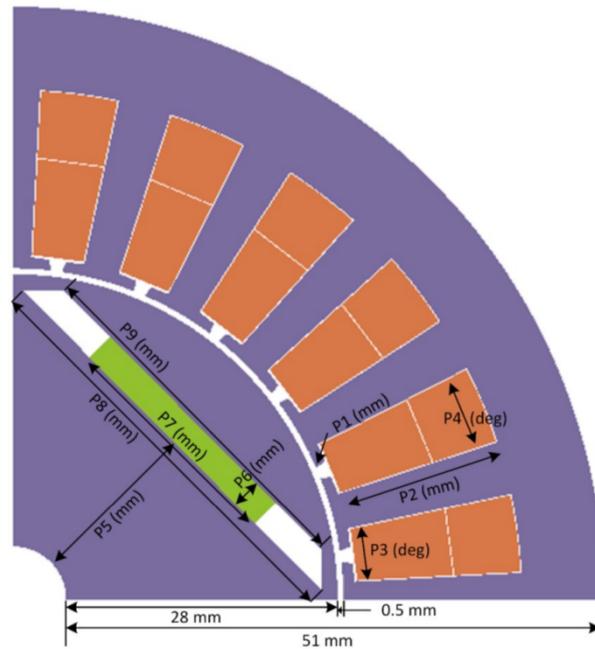


Figura 62 – Elementos do motor de ímã permanente

Variável	Descrição	Intervalo de variação
$X_1$	Dimensão máxima do enrolamento dada por $P_1 + P_2$	$4 \leq X_1 \leq 21.5$
$X_2$	razão entre $P_1$ e $X_1$	$0.01 \leq X_2 \leq 0.4$
$X_3$	mesmo valor de $P_3$	$1 \leq X_3 \leq 14$
$X_4$	mesmo valor de $P_4$	$1 \leq X_4 \leq 14$
$X_5$	mesmo valor de $P_5$	$10 \leq X_5 \leq 24$
$X_6$	Razão entre a espessura (da barra) do ímã permanente e a maior altura possível do rotor. <sup>1</sup>	$0.01 \leq X_6 \leq 1$
$X_7$	Razão entre a largura (da barra) de ímã permanente e a menor largura da ranhura do rotor	$0.01 \leq X_7 \leq 1$
$X_8$	Razão entre a largura inferior da ranhura do rotor e a maior altura possível do rotor <sup>1</sup>	$0.01 \leq X_8 \leq 1$
$X_9$	Razão entre a largura superior da ranhura do rotor e a maior altura possível do rotor <sup>1</sup>	$0.01 \leq X_9 \leq 1$

<sup>1</sup>Os valores da maior altura possível do rotor dependem dos parâmetros  $P_5$  e  $P_6$ .

Tabela 18 – Descrição das variáveis do problema de Otimização do motor de ímã permanente.

apresentadas na tabela 18. A modelagem do problema considerou sete objetivos, que são apresentados na tabela 19. Para tornar mais fácil a visualização dos resultados, foram considerados apenas dois objetivos: a maximização do torque médio (objetivo  $f_1$ ) e a minimização do torque *ripple* (objetivo  $f_2$ ). Para tornar este problema um problema de minimização, multiplicou-se o primeiro objetivo por menos um, de modo que temos o problema de minimização dado pela aplicação  $F(x) = (-f_1, f_2)$ . O resultado obtido é ilustrado na figura 63.

Objetivo	Descrição
$f_1$	Torque médio (Mn)
$f_2$	Torque <i>ripple</i> (%)
$f_3$	Volume do ímã ( $mm^3$ )
$f_4$	Perdas no motor (W)
$f_5$	Perdas no estator (W)
$f_6$	Torque de partida (Nm)
$f_7$	Perdas harmônicas ( <i>ohmic losses</i> ) (W)

Tabela 19 – Objetivos do problema de otimização do motor de ímã permanente.

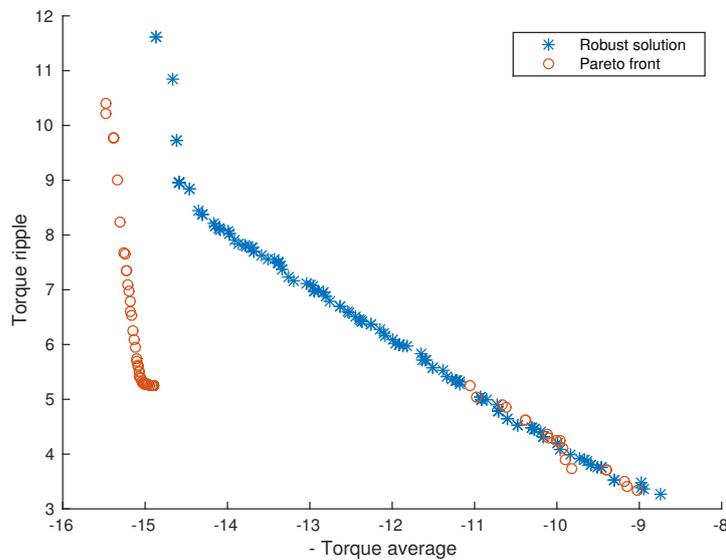


Figura 63 – Fronteira Pareto e solução robusta do problema do motor de ímã permanente

Nesta figura, é apresentado o conjunto de soluções obtidas por meio do algoritmo MOEA/D, representadas por  $\circ$ , e o conjunto de soluções robustas obtidas pelo CR-MOEA/D, representadas por  $*$ . Em uma análise superficial da figura 63, dois fatos se destacam fortemente:

1. As soluções obtidas sem considerar a robustez apresentam uma descontinuidade no intervalo  $[-14.8, -11]$  (valores aproximados) do eixo horizontal (objetivo  $-f_1$ ), enquanto que a Fronteira Pareto Robusta aparenta ser contínua;
2. Uma parte da Fronteira Pareto, localizada na caixa  $[-11, -9] \times [3, 5]$  é robusta, enquanto que a outra parte, localizada na caixa  $[-16, -14.8] \times [5, 12]$ , não é robusta.

Observe que, na região delimitada entre as retas verticais  $-f_1 = -14.8$  e  $-f_1 = -11$ , que não contém pontos da Fronteira Pareto, está ricamente povoada por pontos da Fronteira Pareto Robusta, aumentando as opções do tomador de decisão. Outra característica positiva é a boa dispersão das soluções robustas obtidas.

### 8.3 Considerações Finais

A proposta de algoritmo coevolutivo apresentada nesta seção apresentou bons resultados nos casos analisados, tanto no cenário de otimização geral quanto nas situações onde uma ROI é definida *a priori*. Nos cenários onde foi explorada uma região de interesse não foi detectada a perda de eficiência na busca por soluções robustas em nenhum caso, seja na variação da direção quanto na extensão da ROI. Houve uma degradação na convergência dos algoritmos analisados com o aumento do número de objetivos. Entretanto este é um resultado esperado, sendo observado neste capítulo nos cenários onde foi aplicado um algoritmo de otimização usual (NSGA-III) e na literatura especializada em geral.

Nos estudos preliminares apresentados no artigo [99] ainda não estava disponível a família de funções pdMOP. Em seu lugar foram utilizadas funções destinadas a otimização multiobjetivo usual. Entretanto os resultados obtidos naquele momento são semelhantes aos resultados obtidos neste capítulo: as soluções obtidas são robustas. A grande vantagem dos problemas de otimização propostos neste trabalho são precisão na determinação da região onde as soluções são robustas e a possibilidade de determinar a Fronteira Pareto Robusta do problema, o que permite a análise qualitativa das soluções por meio de métricas de desempenho.

A escolha da abordagem coevolutiva com a presença de duas instâncias de otimização E1 e E2 para minimização do pior caso também nos pareceu adequada. Nas propostas onde a robustez de uma solução candidata é medida por meio de uma amostragem em sua vizinhança o número de avaliações da função objetivo cresce na mesma proporção do tamanho desta amostra. Este cenário leva a uma situação conflitante: a qualidade da estimativa da robustez é proporcional ao tamanho da amostra, mas este tamanho torna o algoritmo computacionalmente muito caro. Se há algum fator que limite o número de avaliações da função objetivo o *trade off* entre o tamanho da amostra e a qualidade da solução obtida pode ser um problema de difícil solução. Por outro lado, na abordagem proposta o número de avaliações da função objetivo é multiplicado por dois devido à presença da instância de otimização adicional E2. Este caso corresponde a um cenário onde é feita uma única amostra na vizinhança da solução candidata nas abordagens que fazem a estimativa da robustez pelo caso médio. Uma única amostra dificilmente seria capaz de produzir uma estimativa razoável da variação da função na solução candidata analisada. Além disso a informação contida nas amostras realizadas é perdida a cada iteração do algoritmo. De forma oposta, na abordagem coevolutiva a população  $\Delta$  de vetores de perturbação evolui juntamente com a população principal  $P$ , reduzindo o efeito da escassez de informação dada por uma única estimativa de pior caso dada pelo vetor  $\delta$ , que converge para o valor ótimo onde seu o impacto é máximo.

Finalmente os resultados obtidos na otimização robusta do motor IPM apresentados no artigo [100] mostrou que esta abordagem é válida em problemas do mundo real.

## 9 Conclusão

Este trabalho reúne alguns resultados sobre otimização multiobjetivo robusta. Por se tratar de um tema específico no contexto da otimização multiobjetivo, este tema não apresenta uma grande variedade de publicações, apesar de sua relevância prática. Esta escassez de técnicas e métodos fez com que muitas ferramentas fossem desenvolvidas neste trabalho visando ao aprimoramento da busca e da tomada de decisão em problemas com a presença de incertezas. Apesar das técnicas desenvolvidas neste trabalho visarem à otimização robusta, todas elas podem ser aplicadas em otimização multiobjetivo usual. A primeira publicação fruto deste trabalho, o algoritmo CR-MOEA/D, foi apresentado no congresso WCCI em 2016 em Vancouver/CA. Naquele momento não havia problemas de *benchmark* de otimização multiobjetivo que consideravam a robustez em todos os objetivos. Em seu lugar foram utilizados problemas usuais (ZDT1 e ZDT2) e problemas onde a robustez era detectada em apenas um dos objetivos (TP2, TP3 e TP4). Todos os problemas abordados contavam com apenas com dois objetivos. Não havia a descrição da Fronteira Pareto Robusta destes problemas, que teve de ser estimada via simulação de Monte Carlo.

Para preencher esta lacuna um novo conjunto de funções de teste foi desenvolvido visando primariamente ter à disposição um conjunto de funções onde os resultados obtidos pelo algoritmo proposto pudessem ser melhor avaliados, principalmente em problemas com mais de dois objetivos. A família de problemas pdMOP desenvolvida mostrou ter uma base sólida e com isto teve sua versatilidade aumentada: foram apresentados juntamente com o problema robusto outros problemas específicos para otimização geral, com liberdade de dimensionamento dos espaços de Decisão e dos Objetivos e diversas outras características desejáveis e até então pouco exploradas por outros problemas de *benchmark*, tais como as restrições de igualdade e desigualdade e a flexibilidade do formato da Fronteira Pareto. A família de problemas robustos que foi proposta também tem sua Fronteira Pareto Robusta conhecida, bem como seu conjunto Pareto ótimo robusto e apresenta total liberdade no escalonamento do número de variáveis e de objetivos. Sua formulação também permite testes em regiões específicas do espaço dos objetivos. Seguindo a ideia básica dos problemas de otimização propostos, também foi apresentada uma nova métrica de avaliação de desempenho dos algoritmos, que independe de uma amostra da Fronteira Pareto.

Uma vez rompido o limite de apenas dois objetivos, a extensão do trabalho para problemas com muitos objetivos foi natural. Entretanto este tipo de problema é, assim como a otimização robusta, um tópico especial de otimização multiobjetivo e uma classe de problemas de difícil solução. A abordagem escolhida foi a de exploração de Regiões de Interesse pois preserva o número de objetivos original. Dadas as características do algoritmo

coevolutivo proposto, um novo método de geração de vetores de peso foi proposto, incluindo a geração de cone de vetores, o que possibilitou a exploração de regiões de interesse. Os experimentos realizados mostraram que o método de geração de vetores de peso proposto é eficiente e que consegue delimitar de forma precisa, fácil e intuitiva uma região específica do espaço dos objetivos. Mais uma vez esta técnica não é restrita a otimização robusta.

Complementando o trabalho desenvolvido também foi apresentado um novo método de visualização de dados em espaços de alta dimensionalidade. Além de todas as virtudes deste método que já foram discutidas em sua seção, este novo método permite a visualização das soluções obtidas e da Fronteira Pareto Robusta em um mesmo gráfico, facilitando a análise dos resultados obtidos.

O último capítulo apresenta uma reunião de todo o trabalho desenvolvido: Um algoritmo foi proposto e aplicado nos problemas de *benchmark* e avaliados na métrica concebida para este trabalho, tanto no cenário de busca de soluções em todo o espaço dos objetivos quanto em regiões de interesse, e as soluções obtidas puderam ser visualizadas com a ferramenta de visualização de dados apresentada. Também foi abordada a otimização robusta de um motor, mostrando que esta técnica não se restringe a problemas puramente teóricos.

## 9.1 Trabalhos futuros

Neste trabalho o foco das atenções foi a busca de soluções de problemas de otimização multiobjetivo com a presença de incertezas em suas variáveis. O ruído considerado tem distribuição uniforme em um intervalo fixo, mas outros tipos de distribuição podem ser considerados de acordo com a especificidade do problema, tais como uma distribuição normal ou binomial. O impacto da presença de incerteza atingiu apenas a convergência das soluções, mas o cenário onde a localização das soluções no espaço dos objetivos sofre a influencia da presença de ruído nas variáveis também é um problema relevante, principalmente no problema de busca por soluções em regiões de interesse.

Expandindo ainda mais o horizonte, outros tipos de problemas com a presença de incerteza podem ser tratados, tais como a otimização ruidosa e a otimização dinâmica. Estes temas apresentam muitos desafios e algumas das técnicas desenvolvidas neste trabalho podem ser adaptadas visando à sua aplicação nestas áreas.

A classe de problemas pdMOP também foi pouco explorada. Uma vez que sua fundamentação está bem definida, é possível fazer sua ampliação para outros tipos de problemas, tais como problemas com a presença de *knee points* ou problemas de otimização com variação temporal por exemplo.

A ideia por trás do CAP-vis é a integração de diferentes métodos de visualização de

---

dados em uma única figura, de modo que a informação perdida em um método possa ser compensada com o uso simultâneo de outra técnica. Neste sentido outros métodos podem ser fundidos com o mesmo princípio, não necessariamente seguindo o *design* circular.

Também há questões de natureza técnica: O método de geração de vetores de peso é computacionalmente caro principalmente em espaços de dimensão elevada e sua eficiência pode ser melhorada.



## Referências

- [1] A. Abraham, L. C. Jain, and R. Goldberg. *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications (Advanced Information and Knowledge Processing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. ISBN 1852337877.
- [2] E. Androulakis, K. Drosou, C. Koukouvinos, and Y.-D. Zhou. Measures of uniformity in experimental designs: A selective overview. *Communications in Statistics - Theory and Methods*, 45(13):3782–3806, may 2016. doi: 10.1080/03610926.2014.966843.
- [3] M. Asafuddoula, T. Ray, and R. Sarker. A decomposition-based evolutionary algorithm for many objective optimization. *IEEE Transactions on Evolutionary Computation*, 19(3):445–460, June 2015. ISSN 1089-778X. doi: 10.1109/TEVC.2014.2339823.
- [4] M. Asafuddoula, H. K. Singh, and T. Ray. Six-sigma robust design optimization using a many-objective decomposition-based evolutionary algorithm. *IEEE Transactions on Evolutionary Computation*, 19(4):490–507, Aug 2015. ISSN 1089-778X. doi: 10.1109/TEVC.2014.2343791.
- [5] T. Back, D. B. Fogel, and Z. Michalewicz, editors. *Handbook of Evolutionary Computation*. IOP Publishing Ltd., Bristol, UK, UK, 1st edition, 1997. ISBN 0750303921.
- [6] M. S. Bazaraa and J. J. Jarvis. *Linear programming an network flows*. John Wiley & Sons, 1977.
- [7] M. S. Bazaraa, H. D. Sheali, and C. M. Shetty. *Nonlinear programming - Theory and Algorithms*. Wiley-Interscience, 1979.
- [8] B. K. Beachkofski and R. V. Grandhi. Improved distributed hypercube sampling. *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Denver, CO*, 2002. doi: <http://dx.doi.org/10.2514/6.2002-1274>.
- [9] S. Bechikh, M. Kessentini, L. B. Said, and K. Ghédira. Preference incorporation in evolutionary multiobjective optimization. In A. R. Hurson, editor, *Advances in Computers*, volume 98 of *Advances in Computers*, pages 141–207. Elsevier, 2015. doi: 10.1016/bs.adcom.2015.03.001.
- [10] S. Bechikh, M. Elarbi, and L. Ben Said. Many-objective optimization using evolutionary algorithms: A survey. In S. Bechikh, R. Datta, and A. Gupta, editors,

- Recent Advances in Evolutionary Multi-objective Optimization*, pages 105–137. Springer International Publishing, Cham, aug 2017. ISBN 978-3-319-42978-6. doi: 10.1007/978-3-319-42978-6\_4.
- [11] A. Ben-Tal, L. E. Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics, 2009.
- [12] B. Bergman, J. de Mare, T. Svensson, and S. Loren. *Robust Design Methodology for Reliability: Exploring the Effects of Variation and Uncertainty*. Wiley, 2009. ISBN 9780470748800.
- [13] H.-G. Beyer and B. Sendhoff. Robust optimization – a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33-34):3190–3218, jul 2007. ISSN 0045-7825. doi: <https://doi.org/10.1016/j.cma.2007.03.003>.
- [14] J. Branke, K. Deb, K. Miettinen, and R. Słowiński, editors. *Multiobjective Optimization*. Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-88908-3.
- [15] R. Cheng, M. Olhofer, and Y. Jin. Reference vector based a posteriori preference articulation for evolutionary multiobjective optimization. In *2015 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, may 2015. doi: 10.1109/cec.2015.7256991.
- [16] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 20(5):773–791, oct 2016. doi: 10.1109/tevc.2016.2519378.
- [17] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, and X. Yao. A benchmark test suite for evolutionary many-objective optimization. *Complex & Intelligent Systems*, 3(1):67–81, mar 2017. doi: 10.1007/s40747-017-0039-7.
- [18] R. Cheng, T. Rodemann, M. Fischer, M. Olhofer, and Y. Jin. Evolutionary many-objective optimization of hybrid electric vehicle control: From general optimization to preference articulation. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 1(2):97–111, apr 2017. doi: 10.1109/tetci.2017.2669104.
- [19] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):256–279, June 2004. ISSN 1089-778X. doi: 10.1109/TEVC.2004.826067.
- [20] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387332545.

- [21] A. M. Cramer, S. D. Sudhoff, and E. L. Zivi. Evolutionary algorithms for minimax problems in robust design. *IEEE Transactions on Evolutionary Computation*, 13(2): 444–453, April 2009. ISSN 1089-778X. doi: 10.1109/TEVC.2008.2004422.
- [22] A. G. Cunha, R. Takahashi, and C. H. Antunes. *Manual de computação evolutiva e metaheurística*. Imprensa da Universidade de Coimbra/UFMG, 2012. ISBN 978-989-26-0583-8 (PDF). doi: 10.14195/978-989-26-0583-8.
- [23] P. Czyżżak and A. Jaskiewicz. Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, 7(1):34–47, jan 1998. ISSN 1099-1360. doi: 10.1002/(sici)1099-1360(199801)7:1<34::aid-mcda161>3.0.co;2-6.
- [24] I. Das and J. E. Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, aug 1998. doi: 10.1137/s1052623496307510.
- [25] L. R. C. de Farias, P. H. M. Braga, H. F. Bassani, and A. F. R. Araújo. MOEA/d with uniformly randomly adaptive weights. In *Proceedings of the Genetic and Evolutionary Computation Conference on - GECCO '18*. ACM Press, 2018. doi: 10.1145/3205455.3205648.
- [26] K. A. De Jong. *Evolutionary Computation: A Unified Approach*. MIT Press, 2006. ISBN 9780262041942.
- [27] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 2001. ISBN 047187339X.
- [28] K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9, 06 2000.
- [29] K. Deb and M. Goyal. A combined genetic adaptive search (geneas) for engineering design. *Computer Science and Informatics*, 26:30–45, 1996.
- [30] K. Deb and H. Gupta. Introducing robustness in multi-objective optimization. *Evol. Comput.*, 14(4):463–494, Dec. 2006. ISSN 1063-6560. doi: 10.1162/evco.2006.14.4.463. URL <http://dx.doi.org/10.1162/evco.2006.14.4.463>.
- [31] K. Deb and H. Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, Aug 2014. ISSN 1089-778X. doi: 10.1109/TEVC.2013.2281535.

- [32] K. Deb and A. Kumar. Light beam search based multi-objective optimization using evolutionary algorithms. In *2007 IEEE Congress on Evolutionary Computation*. IEEE, sep 2007. doi: 10.1109/cec.2007.4424735.
- [33] K. Deb and J. Sundar. Reference point based multi-objective optimization using evolutionary algorithms. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, GECCO '06*, pages 635–642, New York, NY, USA, 2006. ACM. ISBN 1-59593-186-4. doi: 10.1145/1143997.1144112.
- [34] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2): 182–197, Apr 2002. ISSN 1089-778X. doi: 10.1109/4235.996017.
- [35] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization*, pages 105–145. Springer, 2005.
- [36] J. Ferreira, C. M. Fonseca, J. A. Covas, and A. Gaspar-Cunha. *Evolutionary Multi-Objective Robust Optimization*. Tech Education and Publishing, nov 2008. doi: 10.5772/6127.
- [37] A. Flexer. On the use of self-organizing maps for clustering and visualization. In *Principles of Data Mining and Knowledge Discovery*, pages 80–88. Springer Berlin Heidelberg, 1999. doi: 10.1007/978-3-540-48247-5\_9.
- [38] A. Forsgren, P. E. Gill, and M. H. Wright. Interior methods for nonlinear optimization. *SIAM Review*, 44:525–597, 2002.
- [39] A. R. R. Freitas, P. J. Fleming, and F. G. Guimarães. A non-parametric harmony-based objective reduction method for many-objective optimization. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, oct 2013. doi: 10.1109/smc.2013.116.
- [40] A. R. R. Freitas, P. J. Fleming, and F. G. Guimarães. Aggregation trees for visualization and dimension reduction in many-objective optimization. *Information Sciences*, 298:288–314, mar 2015. doi: 10.1016/j.ins.2014.11.044.
- [41] R. Fritzsche. Using parameter-diagrams in automotive engineering. *ATZ worldwide*, 108(6):17–21, 2006. doi: 10.1007/BF03224834.
- [42] A. Gaspar-Cunha and J. A. Covas. Robustness in multi-objective optimization using evolutionary algorithms. *Computational Optimization and Applications*, 39(1):75–96, jun 2007. ISSN 0926-6003. doi: 10.1007/s10589-007-9053-9.

- [43] A. Gaspar-Cunha, J. Ferreira, and G. Recio. Evolutionary robustness analysis for multi-objective optimization: benchmark problems. *Structural and Multidisciplinary Optimization*, 49(5):771–793, 2013. ISSN 1615-1488. doi: 10.1007/s00158-013-1010-x. URL <http://dx.doi.org/10.1007/s00158-013-1010-x>.
- [44] I. Giagkiozis, R. Purshouse, and P. Fleming. Generalized decomposition and cross entropy methods for many-objective optimization. *Information Sciences*, 282:363–387, oct 2014. doi: 10.1016/j.ins.2014.05.045.
- [45] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5):533 – 549, 1986. ISSN 0305-0548. doi: [http://dx.doi.org/10.1016/0305-0548\(86\)90048-1](http://dx.doi.org/10.1016/0305-0548(86)90048-1). URL <http://www.sciencedirect.com/science/article/pii/0305054886900481>. Applications of Integer Programming.
- [46] C. K. Goh, K. C. Tan, C. Y. Cheong, and Y. S. Ong. An investigation on noise-induced features in robust evolutionary multi-objective optimization. *Expert Syst. Appl.*, 37(8):5960–5980, Aug. 2010. ISSN 0957-4174. doi: 10.1016/j.eswa.2010.02.008. URL <http://dx.doi.org/10.1016/j.eswa.2010.02.008>.
- [47] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989. ISBN 0201157675.
- [48] B. L. Gorissen, İ. Yamikoğlu, and D. den Hertog. A practical guide to robust optimization. *Omega*, 53:124–137, jun 2015. ISSN 0305-0483. doi: 10.1016/j.omega.2014.12.006.
- [49] F. Goulart and F. Campelo. Preference-guided evolutionary algorithms for many-objective optimization. *Information Sciences*, 329:236–255, feb 2016. doi: 10.1016/j.ins.2015.09.015.
- [50] Z. Gu, L. Gu, R. Eils, M. Schlesner, and B. Brors. circlize implements and enhances circular visualization in r. *Bioinformatics*, 30(19):2811–2812, jun 2014. doi: 10.1093/bioinformatics/btu393.
- [51] F. Guimarães, D. Lowther, and J. Ramirez. Multiobjective approaches for robust electromagnetic design. *IEEE Transactions on Magnetics*, 42(4):1207–1210, apr 2006. ISSN 0018-9464. doi: 10.1109/tmag.2006.871573.
- [52] F. G. Guimarães, R. M. Palhares, F. Campelo, and H. Igarashi. Design of mixed H<sub>2</sub>/H<sub>∞</sub> control systems using algorithms inspired by the immune system. *Information Sciences*, 177(20):4368–4386, oct 2007. ISSN 0020-0255. doi: <https://doi.org/10.1016/j.ins.2007.02.046>. URL <http://www.sciencedirect.com/science/article/pii/S0020025507001132>.

- [53] F. G. Guimarães, E. F. Wanner, and R. H. Takahashi. A quality metric for multi-objective optimization based on Hierarchical Clustering Techniques. In *2009 IEEE Congress on Evolutionary Computation*. IEEE, may 2009. doi: 10.1109/cec.2009.4983362.
- [54] Y. N. Harari. *Sapiens. Uma Breve História da Humanidade*. L & PM, 2015.
- [55] Z. He and G. G. Yen. Visualization and performance metric in many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 20(3):386–402, jun 2016. doi: 10.1109/tevc.2015.2472283.
- [56] Z. He and G. G. Yen. An improved visualization approach in many-objective optimization. In *2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, jul 2016. doi: 10.1109/cec.2016.7743982.
- [57] J. Hettenhausen, A. Lewis, and S. Mostaghim. Interactive multi-objective particle swarm optimization with heatmap-visualization-based user interface. *Engineering Optimization*, 42(2):119–139, feb 2010. doi: 10.1080/03052150903042632.
- [58] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, and E. Stanley. DNA visual and analytic data mining. In *Proceedings. Visualization' 97 (Cat. No. 97CB36155)*. IEEE, 1997. doi: 10.1109/visual.1997.663916.
- [59] J. H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge, MA, USA, 1992. ISBN 0-262-58111-6.
- [60] S. Huband, P. Hingston, L. Barone, and L. While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, 2006.
- [61] J. Ide and A. Schöbel. Robustness for uncertain multi-objective optimization: a survey and analysis of different concepts. *OR Spectrum*, 38(1):235–271, oct 2015. doi: 10.1007/s00291-015-0418-7.
- [62] Infolytica. Magnet v7 – 2d/3d electromagnetic field simulation software, 2016. URL <http://www.infolytica.com/en/products/magnet/>.
- [63] H. Ishibuchi, K. Doi, H. Masuda, and Y. Nojima. Relation between weight vectors and solutions in MOEA/d. In *2015 IEEE Symposium Series on Computational Intelligence*. IEEE, dec 2015. doi: 10.1109/ssci.2015.127.
- [64] H. Ishibuchi, K. Doi, and Y. Nojima. On the effect of normalization in MOEA/d for multi-objective and many-objective optimization. *Complex & Intelligent Systems*, 3(4):279–294, oct 2017. ISSN 2198-6053. doi: 10.1007/s40747-017-0061-9.

- [65] A. Izmailov and M. Solodov. *Otimização - volume 2 - Métodos Computacionais*. IMPA, 2007.
- [66] L. Jiao, H. Wang, R. Shang, and F. Liu. A co-evolutionary multi-objective optimization algorithm based on direction vectors. *Information Sciences*, 228:90–112, apr 2013. doi: 10.1016/j.ins.2012.12.013.
- [67] Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, 9(1):3–12, oct 2003. doi: 10.1007/s00500-003-0328-5.
- [68] Y. Jin and J. Branke. Evolutionary optimization in uncertain environments – a survey. *Trans. Evol. Comp*, 9(3):303–317, June 2005. ISSN 1089-778X. doi: 10.1109/TEVC.2005.846356. URL <http://dx.doi.org/10.1109/TEVC.2005.846356>.
- [69] Y. Jin and B. Sendhoff. *Trade-Off between Performance and Robustness: An Evolutionary Multiobjective Approach*, chapter Trade-Off between Performance and Robustness: An Evolutionary Multiobjective Approach, pages 237–251. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. ISBN 978-3-540-36970-7. doi: 10.1007/3-540-36970-8\_17.
- [70] E. Kandogan. Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. In *In Proceedings of the IEEE Information Visualization Symposium, Late Breaking Hot Topics*, pages 9–12, 2000.
- [71] Z. S. Kapelan, D. A. Savic, and G. A. Walters. Multiobjective design of water distribution systems under uncertainty. *Water Resources Research*, 41(11), 2005. ISSN 1944-7973. doi: 10.1029/2004WR003787. URL <http://dx.doi.org/10.1029/2004WR003787>. W11407.
- [72] V. Keršulienė, E. K. Zavadskas, and Z. Turskis. SELECTION OF RATIONAL DISPUTE RESOLUTION METHOD BY APPLYING NEW STEP-WISE WEIGHT ASSESSMENT RATIO ANALYSIS (SWARA). *Journal of Business Economics and Management*, 11(2):243–258, jun 2010. doi: 10.3846/jbem.2010.12.
- [73] T. Kohonen. Essentials of the self-organizing map. *Neural Networks*, 37:52–65, jan 2013. doi: 10.1016/j.neunet.2012.09.018.
- [74] R. H. Koochaksaraei. *Qualitative analysis in many-objective optimization with visualization methods*. phdthesis, UFMG, June 2017.
- [75] R. H. Koochaksaraei, R. Enayatifar, and F. G. Guimarães. A New Visualization Tool in Many-Objective Optimization Problems. In *Lecture Notes in Computer Science*, pages 213–224. Springer International Publishing, 2016. doi: 10.1007/978-3-319-32034-2\_18.

- [76] R. H. Koochaksaraei, I. R. Meneghini, V. N. Coelho, and F. G. Guimarães. A new visualization method in many-objective optimization with chord diagram and angular mapping. *Knowledge-Based Systems*, 138:134–154, dec 2017. doi: 10.1016/j.knosys.2017.09.035.
- [77] M. Krzywinski, J. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. A. Marra. Circos: An information aesthetic for comparative genomics. *Genome Research*, 19(9):1639–1645, jun 2009. doi: 10.1101/gr.092759.109.
- [78] D. Kuroiwa and G. M. Lee. On robust multiobjective optimization. *Vietnam Journal of Mathematics*, 40(2&3):305–317, 2012. URL [http://www.math.ac.vn/publications/vjm/VJM\\_40/305-317.htm](http://www.math.ac.vn/publications/vjm/VJM_40/305-317.htm).
- [79] M. Laumanns, L. Thiele, and E. Zitzler. An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research*, 169(3):932–942, mar 2006. doi: 10.1016/j.ejor.2004.08.029.
- [80] H. Li and Q. Zhang. Multiobjective optimization problems with complicated pareto sets, MOEA/d and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302, April 2009. ISSN 1089-778X. doi: 10.1109/tevc.2008.925798.
- [81] K. Li, K. Deb, Q. Zhang, and S. Kwong. An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, 19(5):694–716, oct 2015. ISSN 1089-778X. doi: 10.1109/tevc.2014.2373386.
- [82] K. Li, K. Deb, and X. Yao. Integration of Preferences in Decomposition Multi-Objective Optimization. *ArXiv e-prints*, Jan. 2017.
- [83] M. Li and X. Yao. What weights work for you? adapting weights for any pareto front shape in decomposition-based evolutionary multi-objective optimisation. 09 2017.
- [84] M. Li, F. Guimarães, and D. Lowther. Competitive co-evolutionary algorithm for constrained robust design. *IET Science, Measurement & Technology*, 9:218–223(5), March 2015. ISSN 1751-8822. URL <http://digital-library.theiet.org/content/journals/10.1049/iet-smt.2014.0204>.
- [85] M. Li, L. Zhen, and X. Yao. How to read many-objective solution sets in parallel coordinates [educational forum]. *IEEE Computational Intelligence Magazine*, 12(4): 88–100, nov 2017. doi: 10.1109/mci.2017.2742869.
- [86] A. Lichtenstern. Kriging methods in spatial statistics. Bachelor’s thesis, Technische Universität München, 2013.

- [87] E. L. Lima. *Curso de Análise*, volume 1. IMPA, 2016.
- [88] H. Liu and X. Li. The multiobjective evolutionary algorithm based on determined weight and sub-regional search. In *2009 IEEE Congress on Evolutionary Computation*. IEEE, may 2009. doi: 10.1109/cec.2009.4983176.
- [89] R. Liu, R. Wang, W. Feng, J. Huang, and L. Jiao. Interactive Reference Region Based Multi-Objective Evolutionary Algorithm Through Decomposition. *IEEE Access*, 4:7331–7346, 2016. doi: 10.1109/access.2016.2605759.
- [90] A. López-Jaimes and C. A. C. Coello. Including preferences into a multiobjective evolutionary algorithm to deal with many-objective engineering optimization problems. *Information Sciences*, 277:1–20, sep 2014. doi: 10.1016/j.ins.2014.04.023.
- [91] E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, mar 1963. doi: 10.1175/1520-0469(1963)020<0130:dnf>2.0.co;2.
- [92] H. Lu, M. Zhang, Z. Fei, and K. Mao. Multi-objective energy consumption scheduling based on decomposition algorithm with the non-uniform weight vector. *Applied Soft Computing*, 39:223–239, feb 2016. doi: 10.1016/j.asoc.2015.10.042.
- [93] D. G. Luenberger. *Linear and Nonlinear Programming*. Springer, 2008. doi: 10.1007/978-3-319-18842-3. ISBN: 978-1-4020-7593-3.
- [94] H. Maaranen, K. Miettinen, and A. Penttinen. On initial populations of a genetic algorithm for continuous optimization problems. *Journal of Global Optimization*, 37(3):405–436, jul 2006. ISSN 1573-2916. doi: 10.1007/s10898-006-9056-6. URL <http://dx.doi.org/10.1007/s10898-006-9056-6>.
- [95] S. W. Mahfoud. *Niching methods for genetic algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, 1995.
- [96] K. F. Man, K. S. Tang, and S. Kwong. Genetic algorithms: concepts and applications [in engineering design]. *IEEE Transactions on Industrial Electronics*, 43(5):519–534, Oct 1996. ISSN 0278-0046. doi: 10.1109/41.538609.
- [97] J. R. R. A. Martins and A. B. Lambe. Multidisciplinary design optimization: A survey of architectures. *AIAA Journal*, 51:2049–2075, 2013. doi: 10.2514/1.J051895.
- [98] I. R. Meneghini and F. G. Guimarães. Evolutionary method for weight vector generation in Multi-Objective Evolutionary Algorithms based on decomposition and aggregation. In *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, jun 2017. doi: 10.1109/cec.2017.7969533.

- [99] I. R. Meneghini, F. G. Guimarães, and A. Gaspar-Cunha. Competitive coevolutionary algorithm for robust multi-objective optimization: The worst case minimization. In *2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, jul 2016. doi: 10.1109/cec.2016.7743846.
- [100] I. R. Meneghini, R. C. P. Silva, F. G. Guimarães, and D. Lowther. A coevolutionary method for multi-objective robust optimization. In *IGTE Symposium on Numerical Field Calculation in Electrical Engineering*, Graz, 2016.
- [101] I. R. Meneghini, R. H. Koochaksaraei, F. G. Guimarães, and A. Gaspar-Cunha. Information to the Eye of the Beholder: Data Visualization for Many-Objective Optimization. In *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, jun 2018.
- [102] C. M. Fonseca and P. J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 416–423, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.
- [103] K. Miettinen. *Nonlinear Multiobjective Optimization*. Springer US, 1998. doi: 10.1007/978-1-4615-5563-6.
- [104] K. Miettinen, F. Ruiz, and A. P. Wierzbicki. Introduction to multiobjective optimization: Interactive approaches. In *Multiobjective Optimization*, pages 27–57. Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-88908-3\_2.
- [105] B. L. Miller. *Noise, Sampling, and Efficient Genetic Algorithms*. phdthesis, University of Illinois at Urbana-Champaign, Champaign, IL, USA, 1997. URL <http://hdl.handle.net/2142/81884>. UMI Order No. GAX97-37200.
- [106] S. Mirjalili and A. Lewis. Novel frameworks for creating robust multi-objective benchmark problems. *Inf. Sci.*, 300(C):158–192, Apr. 2015. ISSN 0020-0255. doi: 10.1016/j.ins.2014.12.037. URL <http://dx.doi.org/10.1016/j.ins.2014.12.037>.
- [107] A. Mohammadi, M. N. Omidvar, and X. Li. Reference point based multi-objective optimization through decomposition. In *2012 IEEE Congress on Evolutionary Computation*. IEEE, jun 2012. doi: 10.1109/cec.2012.6256486.
- [108] J. Molina, L. V. Santana, A. G. Hernández-Díaz, C. A. C. Coello, and R. Caballero. g-dominance: Reference point based dominance for multiobjective metaheuristics. *European Journal of Operational Research*, 197(2):685–692, sep 2009. doi: 10.1016/j.ejor.2008.07.015.
- [109] D. C. Montgomery and G. C. Runger. *Estatística Aplicada e Probabilidade Para Engenheiros*. gen/LTC, 2016.

- [110] R. E. Moore, R. B. Kearfott, and M. J. Cloud. *Introduction to Interval Analysis*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2009. ISBN 0898716691, 9780898716696.
- [111] J. Nash. The (dantzig) simplex method for linear programming. *Computing in Science & Engineering*, 2(1):29–31, 2000. ISSN 1521-9615. doi: 10.1109/5992.814654.
- [112] H. Niederreiter. Low-discrepancy and low-dispersion sequences. *Journal of Number Theory*, 30(1):51–70, sep 1988. doi: 10.1016/0022-314x(88)90025-x.
- [113] Y. Nikulin, K. Miettinen, and M. M. Mäkelä. A new achievement scalarizing function based on parameterization in multiobjective optimization. *OR Spectrum*, 34(1): 69–87, aug 2010. doi: 10.1007/s00291-010-0224-1.
- [114] Y.-S. Ong, P. B. Nair, and K. Y. Lum. Max-min surrogate-assisted evolutionary algorithm for robust design. *IEEE Transactions on Evolutionary Computation*, 10(4):392–404, Aug 2006. ISSN 1089-778X. doi: 10.1109/TEVC.2005.859464.
- [115] J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000. ISBN 0-89871-461-3.
- [116] I. Paenke, J. Branke, and Y. Jin. Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation. *IEEE Transactions on Evolutionary Computation*, 10(4):405–420, Aug 2006. ISSN 1089-778X. doi: 10.1109/TEVC.2005.859465.
- [117] I. R. Petersen and R. Tempo. Robust control of uncertain systems: Classical results and recent developments. *Automatica*, 50(5):1315 – 1335, may 2014. ISSN 0005-1098. doi: 10.1016/j.automatica.2014.02.042. URL <http://www.sciencedirect.com/science/article/pii/S0005109814000806>.
- [118] A. Pryke, S. Mostaghim, and A. Nazemi. Heatmap Visualization of Population Based Multi Objective Algorithms. In *Lecture Notes in Computer Science*, pages 361–375. Springer Berlin Heidelberg, 2007. doi: 10.1007/978-3-540-70928-2\_29.
- [119] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu. MOEA/d with adaptive weight adjustment. *Evolutionary Computation*, 22(2):231–264, jun 2014. doi: 10.1162/evco\_a\_00109.
- [120] P. Rakshit, A. Konar, and S. Das. Noisy evolutionary optimization algorithms – a comprehensive survey. *Swarm and Evolutionary Computation*, 33:18–45, apr 2017. doi: 10.1016/j.swevo.2016.09.002.

- [121] S. Rangavajhala, A. Mullur, and A. Messac. The challenge of equality constraints in robust design optimization: examination and new approach. *Structural and Multidisciplinary Optimization*, 34(5):381–401, mar 2007. doi: 10.1007/s00158-007-0104-8.
- [122] S. S. Rao. *Engineering Optimization: Theory and Practice*. John Wiley & Sons, Inc., jun 2009. ISBN 9780470183526. doi: 10.1002/9780470549124.
- [123] C. D. Rosin and R. K. Belew. Methods for competitive co-evolution: Finding opponents worth beating. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 373–381, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1-55860-370-0. URL <http://dl.acm.org/citation.cfm?id=645514.657925>.
- [124] C. D. Rosin and R. K. Belew. New methods for competitive coevolution. *Evol. Comput.*, 5(1):1–29, Mar. 1997. ISSN 1063-6560. doi: 10.1162/evco.1997.5.1.1. URL <http://dx.doi.org/10.1162/evco.1997.5.1.1>.
- [125] M. Rubio-Sanchez, L. Raya, F. Diaz, and A. Sanchez. A comparative study between RadViz and Star Coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):619–628, jan 2016. doi: 10.1109/tvcg.2015.2467324.
- [126] W. Rudin. *Functional Analysis*. International series in pure and applied mathematics. McGraw-Hill, 2 edition edition, 1991. ISBN 9780070542365.
- [127] A. B. Ruiz, R. Saborido, and M. Luque. A preference-based evolutionary algorithm for multiobjective optimization: the weighting achievement scalarizing function genetic algorithm. *Journal of Global Optimization*, 62(1):101–129, jul 2014. doi: 10.1007/s10898-014-0214-y.
- [128] L. B. Said, S. Bechikh, and K. Ghedira. The r-dominance: A new dominance relation for interactive evolutionary multicriteria decision making. *IEEE Transactions on Evolutionary Computation*, 14(5):801–818, oct 2010. doi: 10.1109/tevc.2010.2041060.
- [129] B. Sareni and L. Krahenbuhl. Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation*, 2(3):97–106, Sep 1998. ISSN 1089-778X. doi: 10.1109/4235.735432.
- [130] H. Scheffé. Experiments with mixtures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 20(2):344–360, 1958. ISSN 00359246.
- [131] O. Schutze, A. Lara, and C. A. C. Coello. On the influence of the number of objectives on the hardness of a multiobjective optimization problem. *IEEE Transactions on Evolutionary Computation*, 15(4):444–455, aug 2011. ISSN 1089-778X. doi: 10.1109/tevc.2010.2064321.

- [132] H. Seada and K. Deb. U-NSGA-III: A unified evolutionary optimization procedure for single, multiple, and many objectives: Proof-of-principle results. In *Evolutionary Multi-Criterion Optimization - 8th International Conference, EMO 2015, Guimarães, Portugal, March 29 -April 1, 2015. Proceedings, Part II*, pages 34–49, 2015. doi: 10.1007/978-3-319-15892-1\_3. URL [http://dx.doi.org/10.1007/978-3-319-15892-1\\_3](http://dx.doi.org/10.1007/978-3-319-15892-1_3).
- [133] R. Silva, A. Salimi, M. Li, A. R. R. Freitas, F. G. Guimarães, and D. A. Lowther. Visualization and analysis of tradeoffs in many-objective optimization: A case study on the interior permanent magnet motor design. *IEEE Transactions on Magnetics*, 52(3):1–4, March 2016. ISSN 0018-9464. doi: 10.1109/TMAG.2015.2487979.
- [134] G. L. Soares, F. G. Guimarães, C. A. Maia, J. A. Vasconcelos, and L. Jaulin. Interval robust multi-objective evolutionary algorithm. In *Proceedings of the Eleventh Conference on Congress on Evolutionary Computation, CEC'09*, pages 1637–1643, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-2958-5. URL <http://dl.acm.org/citation.cfm?id=1689599.1689815>.
- [135] D. F. Specht. A general regression neural network. *IEEE Transactions on Neural Networks*, 2(6):568–576, Nov 1991. ISSN 1045-9227. doi: 10.1109/72.97934.
- [136] K. O. Stanley and R. Miikkulainen. Competitive coevolution through evolutionary complexification. *J. Artif. Int. Res.*, 21(1):63–100, Feb. 2004. ISSN 1076-9757. URL <http://dl.acm.org/citation.cfm?id=1622467.1622471>.
- [137] J. Sun, F. Sun, D. Gong, and X. Zeng. A mutation operator guided by preferred regions for set-based many-objective evolutionary optimization. *Complex & Intelligent Systems*, 3(4):265–278, oct 2017. doi: 10.1007/s40747-017-0058-4.
- [138] G. Taguchi. *Introduction to quality engineering: designing quality into products and processes*. Quality Resources, 1986.
- [139] G. Taguchi, S. Chowdhury, and Y. Wu. *Taguchi's Quality Engineering Handbook*. John Wiley & Sons, Inc, 2005.
- [140] E. Talbi. *Metaheuristics: From Design to Implementation*. Wiley Publishing, 2009. ISBN 0470278587, 9780470278581.
- [141] B. Tang. Orthogonal array-based latin hypercubes. *Journal of the American Statistical Association*, 88(424):1392–1397, dec 1993. doi: 10.1080/01621459.1993.10476423.
- [142] B. Tang. Orthogonal array-based latin hypercubes. *Journal of the American Statistical Association*, 88(424):1392–1397, 1993. ISSN 01621459. URL <http://www.jstor.org/stable/2291282>.

- [143] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina. A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary Computation*, 17(3): 411–436, sep 2009. doi: 10.1162/evco.2009.17.3.411.
- [144] Y. Tian, R. Cheng, X. Zhang, and Y. Jin. PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization. *IEEE Computational Intelligence Magazine*, 12(4):73–87, nov 2017. doi: 10.1109/mci.2017.2742868.
- [145] A. Trivedi, D. Srinivasan, K. Pal, and T. Reindl. A MOEA/d with non-uniform weight vector distribution strategy for solving the unit commitment problem in uncertain environment. In *Lecture Notes in Computer Science*, pages 378–390. Springer International Publishing, dec 2016. doi: 10.1007/978-3-319-51691-2\_32.
- [146] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh. A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Transactions on Evolutionary Computation*, 21(3):440–462, June 2017. ISSN 1089-778X. doi: 10.1109/tevc.2016.2608507.
- [147] J. J. Valdes and A. J. Barton. Visualizing high dimensional objective spaces for multi-objective optimization: A virtual reality approach. In *2007 IEEE Congress on Evolutionary Computation*. IEEE, sep 2007. doi: 10.1109/cec.2007.4425019.
- [148] D. A. Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Air Force Institute of Technology, Wright Patterson AFB, OH, USA, 1999. AAI9928483.
- [149] M. Vlachos, C. Domeniconi, D. Gunopulos, G. Kollios, and N. Koudas. Non-linear dimensionality reduction techniques for classification and visualization. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02*. ACM Press, 2002. doi: 10.1145/775047.775143.
- [150] C. von Lüken, B. Barán, and C. Brizuela. A survey on multi-objective evolutionary algorithms for many-objective problems. *Computational Optimization and Applications*, 58(3):707–756, Jul 2014. ISSN 1573-2894. doi: 10.1007/s10589-014-9644-1.
- [151] T. Wagner, N. Beume, and B. Naujoks. Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In *Proceedings of the 4th International Conference on Evolutionary Multi-criterion Optimization, EMO'07*, pages 742–756, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-70927-5. URL <http://dl.acm.org/citation.cfm?id=1762545.1762608>.
- [152] D. J. Walker, R. M. Everson, and J. E. Fieldsend. Visualisation and ordering of many-objective populations. In *IEEE Congress on Evolutionary Computation*. IEEE, jul 2010. doi: 10.1109/cec.2010.5586078.

- [153] D. J. Walker, R. Everson, and J. E. Fieldsend. Visualizing Mutually Nondominating Solution Sets in Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation*, 17(2):165–184, apr 2013. doi: 10.1109/tevc.2012.2225064.
- [154] H. Wang, M. Olhofer, and Y. Jin. A mini-review on preference modeling and articulation in multi-objective optimization: current status and challenges. *Complex & Intelligent Systems*, 3(4):233–245, aug 2017. doi: 10.1007/s40747-017-0053-9.
- [155] R. Wang, J. Xiong, H. Ishibuchi, G. Wu, and T. Zhang. On the effect of reference point in moea/d for multi-objective optimization. *Applied Soft Computing*, 58:25–34, sep 2017. ISSN 1568-4946. doi: <https://doi.org/10.1016/j.asoc.2017.04.002>.
- [156] S. Weinzierl. Introduction to monte carlo methods. *ArXiv High Energy Physics - Phenomenology e-prints*, 2000.
- [157] A. P. Wierzbicki. The use of reference objectives in multiobjective optimization. In *Lecture Notes in Economics and Mathematical Systems*, pages 468–486. Springer Berlin Heidelberg, 1980. doi: 10.1007/978-3-642-48782-8\_32.
- [158] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6): 80, dec 1945. doi: 10.2307/3001968.
- [159] M. Wu, S. Kwong, Y. Jia, K. Li, and Q. Zhang. Adaptive weights generation for decomposition-based multi-objective optimization using gaussian process regression. In *Proceedings of the Genetic and Evolutionary Computation Conference on - GECCO'17*. ACM Press, 2017. doi: 10.1145/3071178.3071339.
- [160] K. Q. Ye. Orthogonal column latin hypercubes and their application in computer experiments. *Journal of the American Statistical Association*, 93(444):1430–1439, dec 1998. doi: 10.1080/01621459.1998.10473803.
- [161] G. Yu, J. Zheng, R. Shen, and M. Li. Decomposing the user-preference in multiobjective optimization. *Soft Computing*, 20(10):4005–4021, jul 2015. doi: 10.1007/s00500-015-1736-z.
- [162] Y. Yuan, Z. Li, and B. Huang. Nonlinear robust optimization with uncertain equality constraints. 2016.
- [163] G. G. Zanabria, L. G. Nonato, and E. Gomez-Nieto. iStar (i\*): An interactive star coordinates approach for high-dimensional data exploration. *Computers & Graphics*, 60:107–118, nov 2016. doi: 10.1016/j.cag.2016.08.007.
- [164] C. Zang, M. I. Friswell, and J. E. Mottershead. A review of robust optimal design and its application in dynamics. *Comput. Struct.*, 83(4-5):315–326, Jan. 2005. ISSN

- 0045-7949. doi: 10.1016/j.compstruc.2004.10.007. URL <http://dx.doi.org/10.1016/j.compstruc.2004.10.007>.
- [165] S. Zapotecas-Martinez, H. E. Aguirre, K. Tanaka, and C. A. C. Coello. On the low-discrepancy sequences and their use in MOEA/d for high-dimensional objective spaces. In *2015 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, may 2015. doi: 10.1109/cec.2015.7257241.
- [166] Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, Dec 2007. ISSN 1089-778X. doi: 10.1109/TEVC.2007.892759.
- [167] Q. Zhang, W. Liu, E. Tsang, and B. Virginas. Expensive multiobjective optimization by moea/d with gaussian process model. *IEEE Transactions on Evolutionary Computation*, 14(3):456–474, June 2010. ISSN 1089-778X. doi: 10.1109/TEVC.2009.2033671.
- [168] A. Zhou and Q. Zhang. A surrogate-assisted evolutionary algorithm for minimax optimization. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–7, July 2010. doi: 10.1109/CEC.2010.5586122.
- [169] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, mar 2011. doi: 10.1016/j.swevo.2011.03.001.
- [170] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms - a comparative case study. In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature, PPSN V*, pages 292–304, London, UK, UK, 1998. Springer-Verlag. ISBN 3-540-65078-4. URL <http://dl.acm.org/citation.cfm?id=645824.668610>.
- [171] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, jun 2000. doi: 10.1162/106365600568202.