

DISSERTAÇÃO DE MESTRADO

**APROXIMAÇÕES ARBITRÁRIAS DE MÓDULO E FASE NA  
MODELAGEM DE SISTEMAS LINEARES INVARIANTES  
AO DESLOCAMENTO COM TÉCNICAS DE  
OTIMIZAÇÃO ‘SOFT-COMPUTING’**

**Flávio Castro Alves Teixeira**

**Brasília, Agosto de 2008**

**UNIVERSIDADE DE BRASÍLIA**

**FACULDADE DE TECNOLOGIA**

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.



UNIVERSIDADE DE BRASÍLIA  
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO

**APROXIMAÇÕES ARBITRÁRIAS DE MÓDULO E FASE NA  
MODELAGEM DE SISTEMAS LINEARES INVARIANTES  
AO DESLOCAMENTO COM TÉCNICAS DE  
OTIMIZAÇÃO ‘SOFT-COMPUTING’**

**Flávio Castro Alves Teixeira**

*Relatório submetido ao Departamento de Engenharia  
Elétrica como requisito parcial para obtenção  
do grau de Mestre em Engenharia Elétrica*

**Banca Examinadora**

Alexandre R. S. Romariz, PhD. ENE/UnB  
*Orientador*

\_\_\_\_\_

Francisco A. de O. Nascimento, Dr. ENE/UnB  
*Examinador interno*

\_\_\_\_\_

Adson Ferreira da Rocha, PhD. ENE/UnB  
*Examinador interno*

\_\_\_\_\_



## FICHA CATALOGRÁFICA

TEIXEIRA, FLÁVIO CASTRO ALVES

Aproximações Arbitrárias de Módulo e Fase na Modelagem de Sistemas Lineares Invariantes ao Deslocamento com Técnicas de Otimização “*Soft-Computing*” [Distrito Federal] 2008.

xvii, 144p., 210 x 297 mm (ENE/FT/UnB, Mestre, Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

1. Processamento Digital de Sinais

2. Otimização Numérica

3. Análise e Projeto de Experimentos Estatísticos

4. Sistemas Inteligentes

I. ENE/FT/UnB

II. Título (série)

## REFERÊNCIA BIBLIOGRÁFICA

TEIXEIRA, F. C. A. (2008). Aproximações Arbitrárias de Módulo e Fase na Modelagem de Sistemas Lineares Invariantes ao Deslocamento com Técnicas de Otimização “*Soft-Computing*”. Dissertação de Mestrado em Engenharia Elétrica, Publicação **PPGENE.DM-350/08**, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 144p.

## CESSÃO DE DIREITOS

AUTOR: Flávio Castro Alves Teixeira.

TÍTULO: Aproximações Arbitrárias de Módulo e Fase na Modelagem de Sistemas Lineares Invariantes ao Deslocamento com Técnicas de Otimização “*Soft-Computing*”.

GRAU: Mestre

ANO: 2008

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

---

Flávio Castro Alves Teixeira

QMSW 04 Lote 10, Ap. 109, Ed. Silco Double Kit, Sudoeste.

70680-400, Brasília – DF – Brasil.



## Agradecimentos

*À minha esposa Luciana, pela paciência, carinho e compreensão nos vários momentos em que não pude dedicar meu tempo à ela.*

*Aos meus pais Flavio e Glória, pela inspiração e exemplo de vida, por me ensinarem a sempre buscar o conhecimento.*

*Aos meus irmãos e familiares pela ajuda e incentivo.*

*Ao meu orientador prof. Dr. Alexandre Romariz, pela paciência, atenção e sabedoria.*

*Aos meus amigos e colegas de trabalho, pelo companheirismo.*

*Flávio Castro Alves Teixeira*



---

## RESUMO

O presente trabalho aborda a modelagem de sistemas lineares invariantes ao deslocamento. A analogia entre sistemas lineares invariantes ao deslocamento e filtros é abordada. O projeto ótimo de filtros digitais é postulado como um problema clássico de aproximação. Como exemplo do problema de aproximação, desenvolve-se detalhadamente no trabalho o problema do projeto ótimo de filtros FIR de fase não-linear com atraso de grupo reduzido nas bandas de passagem. Essa postulação resulta em um problema de otimização não-linear multi-objetivo onde a dificuldade de resolução previne o uso de métodos exatos para a sua resolução. Para abordar esse problema, propõe-se o uso de algoritmos de otimização *soft-computing*, os quais vêm se mostrando na literatura como os mais adequados no presente momento para resolução de problemas complexos e não-lineares. Entretanto, uma problemática no uso desses algoritmos consiste no ajuste ótimo de seus parâmetros internos. Tendo em vista esse problema, desenvolve-se nesse trabalho ferramentas estatísticas de projeto de experimentos e de indicadores de qualidade multi-objetivo, os quais possibilitam obter estimativas de parâmetros ótimos dos algoritmos de otimização propostos. Adicionalmente, aborda-se a questão da comparação de performance entre os algoritmos propostos e a generalização dos resultados para uma classe mais ampla de projetos de filtros e especificações. Experimentos computacionais foram implementados para a geração de filtros ótimos pertencentes ao conjunto Pareto do problema de otimização multi-objetivo. Por fim, considerações a nível do compromisso entre a aproximação simultânea da magnitude e fase do filtro são abordadas, assim como diretrizes para o projetista do filtro na escolha de um filtro que atenda suas expectativas de projeto dentre as várias soluções possíveis para o problema de otimização multi-objetivo do projeto de filtros digitais.

---

## ABSTRACT

This work address the linear time invariant system modeling problem. The analogy between linear time invariant systems and filter is addressed. The optimum design of digital filters is posed as classical approximation problem. The optimum design of FIR filters with non-linear phase on the stop-bands is used as an example of the approximation problem. This postulation results in a multi-objective non-linear optimization problem which prevents using exact methods for the problem resolution. To address this problem, it is proposed the use of soft-computing optimization algorithms, which have been showing best results on complex non-linear problems. Meanwhile, it is desired to obtain best values estimates of the soft-computing internal parameters. For this problem, it is developed statistical tools based on experimental design and multi-objective quality indicators to obtain best parameters values estimates of the soft-computing algorithms internal parameters. Additionally, the algorithms performance comparison is addressed in conjunction with generalizations of the results to a wide range of problems and filter specifications. Computational experiments were implemented to generate optimum filters that belong to the Pareto set of the multi-objective problem under study. Finally, considerations involving the trade-off between the simultaneous approximation of both filter magnitude and phase is addressed, as well as guidelines to the filter designer on choosing the best filter that meets its expectations.



# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>1</b>
<b>2</b>	<b>PROJETO DE FILTROS DIGITAIS FIR .....</b>	<b>5</b>
2.1	INTRODUÇÃO .....	5
2.2	PROJETO DE FILTROS FIR COM FASE LINEAR .....	9
2.2.1	FORMA DOS COEFICIENTES.....	9
2.2.2	PROJETO POR AMOSTRAGEM NA FREQUÊNCIA .....	10
2.2.3	PROJETO POR APROXIMAÇÃO COM FUNÇÕES-JANELA .....	11
2.2.4	PROJETO DE APROXIMAÇÃO POR OTIMIZAÇÃO.....	13
2.3	PROJETO DE FILTROS FIR COM FASE NÃO-LINEAR - APROXIMAÇÕES ARBITRÁRIAS DE MÓDULO E FASE.....	16
2.3.1	INTRODUÇÃO .....	16
2.3.2	PROBLEMA CLÁSSICO DE APROXIMAÇÃO .....	17
2.3.3	FORMA DA FUNÇÃO TRANSFERÊNCIA E DA RESPOSTA EM FREQUÊNCIA .	18
2.3.4	FUNÇÃO OBJETIVO PARA OTIMIZAÇÃO .....	20
<b>3</b>	<b>TEORIA DE OTIMIZAÇÃO APLICADA AO PROJETO DE FILTROS DIGI- TAIS .....</b>	<b>23</b>
3.1	INTRODUÇÃO .....	23
3.2	DEFINIÇÃO DO PROBLEMA DE OTIMIZAÇÃO .....	26
3.3	CLASSIFICAÇÃO DOS PROBLEMAS DE OTIMIZAÇÃO.....	27
3.3.1	OTIMIZAÇÃO SEM RESTRIÇÃO .....	28
3.3.2	OTIMIZAÇÃO COM RESTRIÇÃO .....	28
3.3.3	OTIMIZAÇÃO MONO-OBJETIVO .....	31
3.3.4	OTIMIZAÇÃO MULTI-OBJETIVO .....	31
3.4	ANÁLISE DO PROBLEMA DE OTIMIZAÇÃO NO PROJETO DE FILTROS FIR DE FASE LINEAR .....	36
3.4.1	FILTROS COM CRITÉRIO DOS MÍNIMOS QUADRADOS .....	37
3.4.2	FILTROS COM CRITÉRIO MINIMAX OU CHEBYSHEV .....	38
3.5	ANÁLISE DO PROBLEMA DE OTIMIZAÇÃO NO PROJETO DE FILTROS FIR DE FASE NÃO-LINEAR.....	39
<b>4</b>	<b>ALGORITMOS DE OTIMIZAÇÃO NÃO-LINEAR BASEADOS NO GRADI- ENTE .....</b>	<b>43</b>
4.1	INTRODUÇÃO .....	43
4.2	SOLUÇÕES LOCAIS E CONDIÇÕES DE PRIMEIRA E SEGUNDA ORDEM ....	46
4.3	ALGORITMOS DE BUSCA EM LINHA.....	48

4.4	MÉTODO DO GRADIENTE MÁXIMO .....	51
4.5	MÉTODOS COM GRADIENTE CONJUGADO .....	52
4.6	MÉTODO DE NEWTON .....	54
4.7	MÉTODO QUASI-NEWTON .....	56
<b>5</b>	<b>ALGORITMOS DE OTIMIZAÇÃO SOFT-COMPUTING.....</b>	<b>61</b>
5.1	INTRODUÇÃO .....	61
5.2	INICIALIZAÇÃO DA POPULAÇÃO .....	62
5.3	FUNÇÃO DE ADEQUAÇÃO .....	63
5.4	ALGORITMOS GENÉTICOS .....	65
5.4.1	CODIFICAÇÃO DAS SOLUÇÕES CANDIDATAS.....	66
5.4.2	OPERADOR DE SELEÇÃO .....	67
5.4.3	OPERADORES DE REPRODUÇÃO.....	68
5.5	RESFRIAMENTO SIMULADO .....	71
5.5.1	RESFRIAMENTO BOLTZMANN .....	72
5.6	OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS .....	73
5.6.1	FÓRMULAS DE ATUALIZAÇÃO DA VELOCIDADE E POSIÇÃO DAS PARTÍCULAS.....	73
5.6.2	TOPOLOGIA DA VIZINHANÇA DAS PARTÍCULAS .....	75
<b>6</b>	<b>ANÁLISE EXPLORATÓRIA ESTATÍSTICA DOS ALGORITMOS SOFT-COMPUTING 77</b>	
6.1	INTRODUÇÃO .....	77
6.2	ANÁLISE EXPLORATÓRIA ESTATÍSTICA EM PROBLEMAS DE OTIMIZAÇÃO MONO-OBJETIVO .....	78
6.3	INDICADORES DE QUALIDADE EM PROBLEMAS MULTI-OBJETIVO .....	82
6.4	EXTENSÕES MULTI-OBJETIVO PARA A ANÁLISE EXPLORATÓRIA ESTATÍSTICA .....	85
6.4.1	ESTIMAÇÃO DOS PARÂMETROS ÓTIMOS DOS ALGORITMOS DE OTIMIZAÇÃO .....	85
6.4.2	COMPARAÇÃO DE PERFORMANCE ENTRE OS ALGORITMOS DE OTIMIZAÇÃO .....	88
<b>7</b>	<b>RESULTADOS .....</b>	<b>91</b>
7.1	INTRODUÇÃO .....	91
7.2	FERRAMENTAS COMPUTACIONAIS DE SUPORTE .....	92
7.2.1	FRAMEWORK DE OTIMIZAÇÃO .....	92
7.2.2	ANÁLISE EXPLORATÓRIA ESTATÍSTICA .....	94
7.2.3	ANÁLISE GRÁFICA DOS FILTROS RESULTANTES.....	95
7.3	RESULTADOS DA ANÁLISE EXPLORATÓRIA ESTATÍSTICA .....	95
7.3.1	RESULTADOS DA ESTIMAÇÃO DOS PARÂMETROS ÓTIMOS DOS ALGORITMOS .....	97

7.3.2	RESULTADOS DA COMPARAÇÃO DE PERFORMANCE DOS ALGORITMOS DE OTIMIZAÇÃO .....	101
7.4	CONSIDERAÇÕES ADICIONAIS EM PROJETOS DE FILTROS COM A METODOLOGIA DE OTIMIZAÇÃO MULTI-OBJETIVO .....	102
7.4.1	PROJETO DE FILTRO FIR REJEITA-FAIXA COM ATRASO DE GRUPO REDUZIDO NAS BANDAS DE PASSAGEM .....	102
7.4.2	PROJETO DE FILTRO FIR PASSA-FAIXA COM ATRASO DE GRUPO REDUZIDO NA BANDA DE PASSAGEM .....	103
<b>8</b>	<b>CONCLUSÕES .....</b>	<b>121</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>123</b>
	<b>ANEXOS.....</b>	<b>129</b>
<b>I</b>	<b>ALGORITMOS.....</b>	<b>131</b>
I.1	ALGORITMOS DE OTIMIZAÇÃO NÃO-LINEAR BASEADOS NO GRADIENTE ..	131
I.1.1	MÉTODO QUASI-NEWTON .....	131
I.1.2	ALGORITMO DE BUSCA EM LINHA.....	131
I.2	ALGORITMOS DE OTIMIZAÇÃO SOFT-COMPUTING .....	134
I.2.1	ALGORITMO GENÉTICO .....	134
I.2.2	RESFRIAMENTO SIMULADO .....	138
I.2.3	OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS .....	139
<b>II</b>	<b>FRAMEWORK DE OTIMIZAÇÃO .....</b>	<b>141</b>
II.1	EXEMPLOS DE ARQUIVOS DE CONFIGURAÇÃO DO FRAMEWORK DE OTIMIZAÇÃO .....	141



# LISTA DE FIGURAS

2.1	Filtro passa-baixa ideal.....	6
2.2	Filtro passa-alta ideal .....	6
2.3	Filtro passa-faixa ideal .....	7
2.4	Filtro rejeita-faixa ideal .....	8
3.1	Exemplo de função linear .....	24
3.2	Exemplo de função quadrática .....	25
3.3	Exemplo de função não-linear.....	25
3.4	Exemplo dos conjuntos $S$ e $F$ .....	32
3.5	Dualidade em problemas não-convexos.....	36
4.1	Linha $\ell$ com direção de busca $p$ em duas dimensões .....	44
6.1	Exemplo de diagrama de pontos para o algoritmo genético.....	87
6.2	Exemplo de superfície de resposta para algoritmo genético [29] .....	89
7.1	Diagrama de classes dos algoritmos de otimização .....	105
7.2	Diagrama de classes do projeto de filtros FIR .....	106
7.3	Diagrama de classes dos experimentos estatísticos .....	107
7.4	Diagrama de classes de configuração do framework de otimização .....	108
7.5	Especificação da resposta em frequência do filtro rejeita-faixa para análise exploratória estatística .....	109
7.6	Diagrama de Pontos do algoritmo genético com limiar de 0.94 .....	110
7.7	Processo de escolha do valor limiar no diagrama de pontos .....	111
7.8	Superfície de resposta do experimento de estimação de parâmetros ótimos do algoritmo genético .....	112
7.9	Diagrama de Pontos da otimização por enxame de partículas com limiar de 0.9 .....	112
7.10	Superfície de resposta do experimento de estimação de parâmetros ótimos da otimização por enxame de partículas.....	113
7.11	Comparativo do conjunto Pareto ótimo do projeto de filtro FIR rejeita-faixa .....	113
7.12	Comparativo para diferentes filtros do conjunto Pareto-ótimo obtidos com o algoritmo Quasi-Newton .....	114
7.13	Comparativo para diferentes filtros do conjunto Pareto-ótimo obtidos com o algoritmo otimização por enxame de partículas .....	115
7.14	Especificação da resposta em frequência do filtro passa-faixa .....	116
7.15	Comparativo do conjunto Pareto ótimo do projeto de filtro FIR passa-faixa .....	117
7.16	Comparativo para diferentes filtros do conjunto Pareto-ótimo obtidos com o algoritmo genético .....	118



7.17 Comparativo para diferentes filtros do conjunto Pareto-ótimo obtidos com o resfriamento simulado.....	119
--	-----

# LISTA DE TABELAS

6.1	Experimento completamente aleatorizado.....	80
6.2	Experimento em blocos completamente aleatorizados para o algoritmo genético....	86
6.3	Experimento em blocos completamente aleatorizados para a otimização por enxame de partículas.....	86
6.4	Experimento em blocos completamente aleatorizados para comparação dos algoritmos de otimização.....	89
7.1	Finalização dos intervalos para o algoritmo genético.....	97
7.2	Tabela ANOVA com somatório particionado dos quadrados do experimento com algoritmo genético .....	98
7.3	Finalização dos intervalos para a otimização por enxame de partículas .....	99
7.4	Tabela ANOVA com somatório particionado dos quadrados do experimento com a otimização por enxame de partículas .....	100
7.5	Tabela ANOVA para o experimento de comparação dos algoritmos.....	101
7.6	Estatísticas de $I_{\epsilon}(\mathbf{Z}^*, \mathbf{A})$ para o experimento comparativo de performance .....	101



# LISTA DE SÍMBOLOS

## Siglas

SLID	Sistema Linear Invariante ao Deslocamento
FIR	Resposta ao Impulso Finita
IIR	Resposta ao Impulso Infinita
RS	Resfriamento Simulado
OEA	Otimização por Enxame de Partículas
ANOVA	Análise da Variância



# 1 INTRODUÇÃO

O presente trabalho propõe o estudo e desenvolvimento de técnicas na modelagem de sistemas lineares invariantes ao deslocamento. Primeiramente, o presente trabalho concentra-se na modelagem de sistemas que realizam transformações em sinais em tempo discreto. Embora os sinais possam ser representados de várias maneiras, geralmente utiliza-se uma representação matemática, descrevendo um sinal como uma função de uma ou mais variáveis independentes [1]. Uma convenção comum na área de processamento de sinais é a de considerar a variável independente como o tempo na representação matemática de um sinal, ex: sinal de voz, embora em exemplos específicos a variável independente possa de fato não representar o tempo, ex: sinal de uma imagem fotográfica representado como a função de intensidade luminosa de duas variáveis espaciais. [1].

O conceito de um sistema discreto no tempo é definido matematicamente como uma transformação ou operador  $T\{\cdot\}$  que mapeia uma seqüência (sinal discreto no tempo) de entrada com valores  $x(n)$  em uma seqüência de saída com valores  $y(n)$ , ou seja [1]:

$$y(n) = T\{x(n)\} \quad (1.1)$$

A primeira propriedade dos sistemas que serão abordados no presente trabalho é a de linearidade. A classe de sistemas lineares é definida pelo princípio da superposição. Faça  $y_1(n)$  e  $y_2(n)$  serem as respostas de um sistema quando  $x_1(n)$  e  $x_2(n)$  são suas respectivas entradas. O sistema em questão é linear se e somente se [1]:

$$T\{ax_1(n) + bx_2(n)\} = aT\{x_1(n)\} + bT\{x_2(n)\}, \quad (1.2)$$

para duas constantes arbitrárias  $a$  e  $b$ .

A segunda propriedade dos sistemas abordados é a de invariância ao deslocamento (invariância no tempo). Nesses sistemas, um deslocamento no tempo ou atraso na seqüência de entrada causa um deslocamento correspondente na seqüência de saída [1]. Ou seja, para uma seqüência de entrada  $x(n)$  e qualquer inteiro  $n_0$  tem-se que:

$$T\{x(n - n_0)\} = y(n - n_0) \quad (1.3)$$

Uma possível forma de caracterização (no domínio do tempo) dos sistemas lineares invariantes ao deslocamento (SLID) consiste na sua representação pela resposta ao impulso. Especificamente, considere  $h(n)$  como a resposta do sistema para um impulso ocorrendo em  $n = k$ ,  $\delta(n - k)$ . Em [1] mostra-se que a partir das propriedades de linearidade (1.2) e de invariância ao deslocamento (1.3), tem-se que a saída de um SLID é definida por:

$$y(n) = \sum_{k=-\infty}^{\infty} x(k) h(n-k) \quad (1.4)$$

A relação em (1.4) é também chamada de soma da convolução.

Outra forma possível de caracterização dos SLID's consiste na sua representação no domínio da frequência  $H(e^{j\omega})$  em termos de senóides e exponenciais complexas. Ou seja, a saída  $y(n)$  de um SLID no domínio da frequência fica [1]:

$$y(n) = H(e^{j\omega}) e^{j\omega n}, \quad (1.5)$$

onde  $H(e^{j\omega})$ , ou a resposta em frequência do sistema, é definida por:

$$H(e^{j\omega}) = \sum_{k=-\infty}^{\infty} h(k) e^{-j\omega k} \quad (1.6)$$

Embora o presente trabalho envolva diretamente a questão da modelagem de SLID's, escolheu-se, particularmente, em focar no desenvolvimento de técnicas de projeto de filtros digitais. O termo filtro é comumente utilizado para descrever um dispositivo que discrimina, de acordo com alguns atributos dos objetos aplicados na sua entrada, o que passa através dele. Um SLID executa um tipo de discriminação ou filtragem entre os vários componentes de frequência na sua entrada [2]. A natureza dessa ação de filtragem é determinada pelas características da resposta em frequência  $H(e^{j\omega})$ , as quais, por sua vez, dependem na escolha dos parâmetros do sistema, conforme será visto no Capítulo 2. Dessa forma, ajustando esses parâmetros, pode-se projetar filtros seletivos em frequência que permitem passar certos componentes de frequência do sinal de interesse em algumas bandas, enquanto que em outras bandas os componentes restantes do sinal sejam atenuados [2].

Em geral, um SLID modifica o espectro de um sinal de entrada  $X(e^{j\omega})$  de acordo com sua resposta em frequência  $H(e^{j\omega})$ , resultando em um sinal de saída  $Y(e^{j\omega})$  com espectro  $Y(e^{j\omega}) = H(e^{j\omega}) X(e^{j\omega})$ . De certa forma,  $H(e^{j\omega})$  atua como uma função de ponderação ou função de modelagem espectral aos diferentes componentes em frequência do sinal de entrada [2]. Quando visto nesse contexto, qualquer SLID pode ser considerado como um filtro, embora esse sistema não necessariamente bloqueie completamente alguns ou todos os componentes em frequência do sinal de entrada. Consequentemente, de forma geral, os termos “sistema linear invariante ao deslocamento” e “filtro” são sinônimos e geralmente utilizados indiferentemente [2]. Dessa forma, as técnicas de projeto de filtros que serão desenvolvidas no presente trabalho são diretamente aplicáveis na modelagem de sistemas lineares invariantes ao deslocamento.

Tendo feito a analogia entre SLID e filtros, o presente trabalho propõe então, o emprego de diferentes algoritmos de otimização aplicados a um problema específico de otimização na área de processamento digital de sinais. Mais especificamente, o problema envolve o projeto de filtros

digitais, onde deseja-se aproximar simultaneamente a magnitude e fase da resposta em frequência. Tradicionalmente, grande parte dos filtros digitais projetados na prática são especificados em termos de restrições na magnitude da resposta em frequência apenas, embora nenhuma restrição na fase da resposta em frequência seja imposta. Entretanto, conforme será visto no Capítulo 2, em determinadas aplicações especializadas deseja-se aproximar simultaneamente ambas magnitude e fase da resposta em frequência. Uma das aplicações que será abordada consiste no projeto ótimo de filtros digitais FIR com fase não-linear, onde deseja-se aproximar simultaneamente a magnitude da resposta em frequência e o atraso de grupo do filtro. Esse problema específico será postulado como um problema de otimização multi-objetivo não-linear, onde uma série de algoritmos de otimização propostos na literatura podem ser utilizados.

Uma classe importante de algoritmos de otimização foram propostos numa área denominada de *soft-computing*, a qual envolve o estudo de algoritmos evolucionários, redes neurais, lógica nebulosa, entre outras técnicas diversas. Uma problemática no emprego desses algoritmos envolve a seleção de forma ótima de seus parâmetros internos. O presente trabalho propõe então o uso de uma análise exploratória estatística que permita obter estimativas de parâmetros internos ótimos dos algoritmos abordados. Adicionalmente, considerações a nível da performance com a análise estatística dos diferentes algoritmos de otimização empregados no projeto de filtros digitais FIR serão detalhadamente expostas.

Com a metodologia de projeto ótimo de filtros digitais propostas, serão apresentados alguns resultados com relação a aplicação de diferentes algoritmos de otimização ao problema proposto, assim como considerações sobre os filtros projetados e o compromisso existente entre a aproximação simultânea da magnitude e do atraso de grupo do filtro. Considerações sobre a generalização dos resultados para uma classe mais ampla de especificações e problemas de otimização no projeto de filtros digitais também será abordado.

O presente trabalho é organizado da seguinte forma. O Capítulo 2 apresenta discussão detalhada do projeto de filtros digitais. Mais especificamente, as diversas técnicas de projeto serão explanadas. Adicionalmente, o projeto de filtros digitais FIR de fase não-linear será postulado como um problema de otimização não-linear multi-objetivo.

O Capítulo 3 apresenta teoria de otimização aplicado ao projeto de filtros digitais. As diversas subclasses como otimização linear, não-linear, com restrição e sem restrição serão detalhadamente expostas. Adicionalmente análises do problema de projeto de filtros digitais sobre a óptica da teoria de otimização serão abordadas.

O Capítulo 4 apresenta uma classe importante de algoritmos de otimização. Essa classe envolve algoritmos que fazem uso do gradiente da função objetivo para geração da nova direção de busca no processo de otimização. Entre os algoritmos abordados, encontra-se o algoritmo do gradiente máximo, gradiente conjugado, Newton e algoritmo Quasi-Newton.

O Capítulo 5 apresenta discussão detalhada dos algoritmos de otimização *soft-computing* que serão empregados no problema de otimização multi-objetivo do projeto de filtros digitais FIR de



fase não-linear. Entre os algoritmos propostos encontram-se o algoritmo genético, resfriamento simulado e otimização por enxame de partículas.

O Capítulo 6 apresenta discussão detalhada da análise exploratória estatística que será utilizada para estimação dos parâmetros ótimos dos algoritmos de otimização *soft-computing*, assim como a criação de experimentos estatísticos que possibilitem a comparação de performance entre os diversos algoritmos de otimização propostos para o problema de otimização multi-objetivo no projeto de filtros digitais.

Por fim o Capítulo 7 apresenta resultados dos algoritmos de otimização aplicados no projeto de vários filtros digitais, assim como considerações de compromisso entre a abordagem de aproximação simultânea da magnitude e fase no projeto de filtros digitais.

# 2 PROJETO DE FILTROS DIGITAIS FIR

## 2.1 INTRODUÇÃO

No sentido estrito, o termo filtro seletivo em frequência sugere um sistema que passa certos componentes em frequência e rejeita totalmente todos os outros, mas num contexto mais amplo, qualquer sistema que modifica certas frequências em relação às outras é também classificado como um filtro [1].

Um filtro digital pode ser identificado de forma única no domínio tempo/espço por sua resposta ao impulso  $h(n)$ . Alternativamente, um filtro digital pode ser caracterizado de forma única no domínio da frequência por sua resposta em frequência  $H(e^{j\omega})$ , também expressa pela DTFT, transformada de Fourier em tempo discreto, da sequência  $h(n)$  [3].

Os filtros digitais podem ser classificados em dois tipos principais:

- FIR (*Finite Impulse Response*), Resposta ao Impulso Finita também chamados de não-recursivos.
- IIR (*Infinite Impulse Response*), Resposta ao Impulso Infinita ou também chamados de recursivos.

Nos filtros FIR,  $h(n)$  é diferente de zero somente para um número finito de amostras enquanto que nos filtros IIR,  $h(n)$  possui um numero infinito de amostras diferente de zero. Adicionalmente, no caso dos filtro FIR, as amostras da sequência  $h(n)$  são também chamadas de coeficientes do filtro. No caso dos filtros IIR, os coeficientes do filtro incluem termos de realimentação na equação de diferenças [3].

A área de projeto de filtros digitais tem sido estudada extensivamente durante os últimos 40 anos. O projeto e realização de filtros digitais envolve a combinação de teoria, aplicações e tecnologias [3]. Comumente, no projeto de filtros, as especificações são dadas ou no domínio da frequência pela especificação da resposta em frequência desejada  $D(e^{j\omega})$ , ou no domínio do tempo pela resposta ao impulso  $h(n)$  ou pela função de transferência  $H(z)$ . É importante notar que, para as especificações no domínio da frequência, a resposta desejada  $D(e^{j\omega})$  é em geral uma variável complexa, consistindo da magnitude da resposta desejada  $|D(e^{j\omega})|$  e da resposta em fase desejada  $\angle D(e^{j\omega})$ . O foco desse trabalho, conforme será desenvolvido nesse capítulo, consiste então de se aproximar, desejavelmente no sentido ótimo, simultaneamente  $|D(e^{j\omega})|$  e  $\angle D(e^{j\omega})$ .

Como a resposta em frequência de um filtro digital é sempre periódica na variável de frequência  $\omega$  com período  $2\pi$ , as especificações de projeto do filtro necessitam ser definidas apenas para um período, geralmente na região de frequência  $[-\pi, \pi]$ . Além disso, quando a resposta em

freqüência é conjugada simétrica,  $D^*(e^{j\omega}) = D(e^{-j\omega})$ , então é suficiente especificar a resposta apenas no intervalo positivo de freqüências  $[0, \pi]$  [3]. Esse caso conjugado-simétrico da resposta em freqüência é o mais comum, já que ele corresponde a filtros digitais com coeficientes reais. Nesse trabalho serão abordados filtros digitais apenas com coeficientes reais.

Os casos mais simples de especificação da magnitude da resposta em freqüência de filtros digitais consistem nas respostas ideais passa-baixa, passa-alta, passa-faixa e rejeita-faixa.

O filtro passa-baixa ideal possui a magnitude da resposta em freqüência dada por:

$$|H(e^{j\omega})| = \begin{cases} 1, & \text{para } \omega \leq \omega_c \\ 0, & \text{para } \omega_c \leq \omega \leq \pi \end{cases} \quad (2.1)$$

A Figura 2.1 ilustra as especificações dadas em (2.1). Como o nome diz, esse tipo de filtro permite passar apenas as baixas freqüências, ex: freqüências abaixo da freqüência de corte  $\omega_c$ .

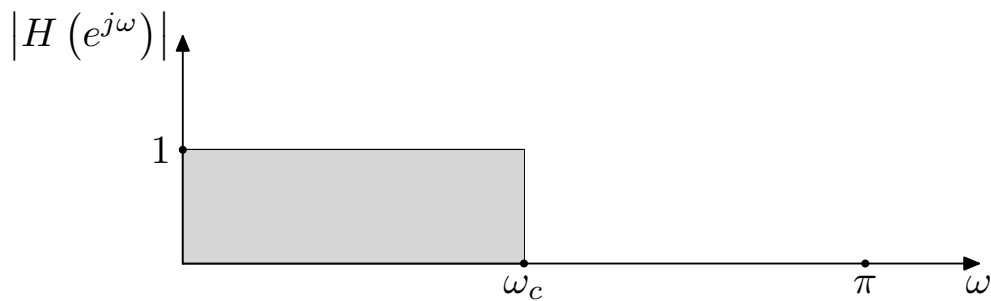


Figura 2.1: Filtro passa-baixa ideal

Por sua vez, o filtro passa-alta ideal possui a magnitude da resposta em freqüência dada por:

$$|H(e^{j\omega})| = \begin{cases} 0, & \text{para } \omega \leq \omega_c \\ 1, & \text{para } \omega_c \leq \omega \leq \pi \end{cases} \quad (2.2)$$

A Figura 2.2 ilustra as especificações dadas em (2.2). De forma análoga, esse tipo de filtro permite passar apenas as altas freqüências, ex: freqüências acima da freqüência de corte  $\omega_c$ .

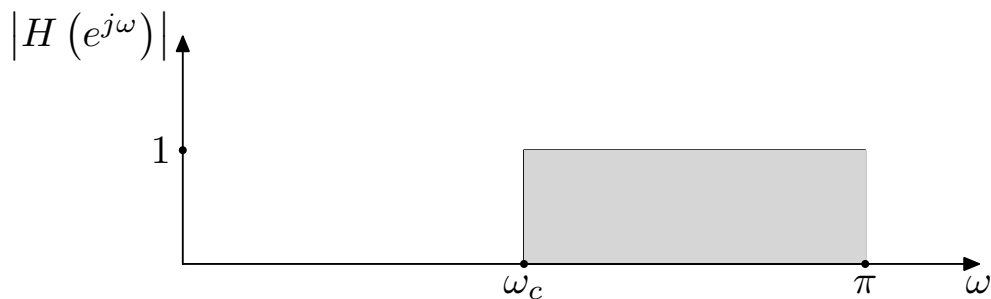


Figura 2.2: Filtro passa-alta ideal

Já o filtro passa-faixa ideal possui a magnitude da resposta em freqüência dada por:

$$|H(e^{j\omega})| = \begin{cases} 0, & \text{para } 0 \leq \omega \leq \omega_{c1} \\ 1, & \text{para } \omega_{c1} \leq \omega \leq \omega_{c2} \\ 0, & \text{para } \omega_{c2} \leq \omega \leq \pi \end{cases} \quad (2.3)$$

A Figura 2.3 ilustra as especificações dadas em (2.3). Esse tipo de filtro permite passar apenas as frequências na faixa de passagem  $\omega_{c1} \leq \omega \leq \omega_{c2}$

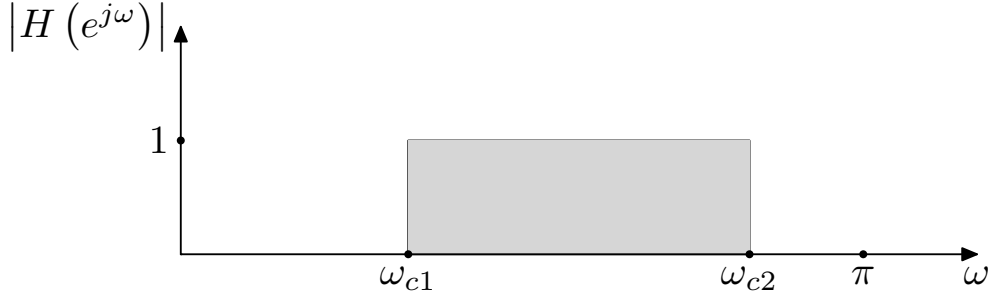


Figura 2.3: Filtro passa-faixa ideal

Por fim, o filtro rejeita-faixa ideal possui a magnitude da resposta em frequência dada por:

$$|H(e^{j\omega})| = \begin{cases} 1, & \text{para } 0 \leq \omega \leq \omega_{c1} \\ 0, & \text{para } \omega_{c1} \leq \omega \leq \omega_{c2} \\ 1, & \text{para } \omega_{c2} \leq \omega \leq \pi \end{cases} \quad (2.4)$$

A Figura 2.4 ilustra as especificações dadas em (2.4). Esse tipo de filtro rejeita todas as frequências na faixa de rejeição  $\omega_{c1} \leq \omega \leq \omega_{c2}$ .

Além dos filtros ideais padrões citados acima, adicionalmente, outros dois tipos de filtro digitais bastante comuns na área de processamento digital de sinais consistem dos diferenciadores e dos transformadores Hilbert [4], embora os mesmos não sejam abordados nesse trabalho.

Os filtros ideais apresentados possuem descontinuidades na magnitude da resposta em frequência e não podem ser implementados diretamente. Eles devem ser aproximados com um sistema fisicamente realizável, ex: a descontinuidade na frequência de corte  $\omega_c$  no filtro passa-baixa da figura 2.1 deve ser substituída por uma faixa (banda) de transição na qual a magnitude muda suavemente da faixa de passagem  $\omega \leq \omega_c$  para a faixa de rejeição  $\omega \geq \omega_c$ .

Tradicionalmente, grande parte dos filtros digitais projetados na prática são especificados em termos de restrições na magnitude da resposta em frequência, embora nenhuma restrição na fase da resposta em frequência seja imposta além das restrições implícitas de estabilidade e/ou causalidade, ex: pólos dentro do círculo unitário para filtros IIR e fase linear da resposta em frequência para filtros FIR [3]. Por exemplo, para filtros FIR, geralmente é estabelecido fase linear (inclinação constante) da resposta em frequência do filtro, ou seja:

$$\Theta(\omega) = -M\omega, \quad (2.5)$$

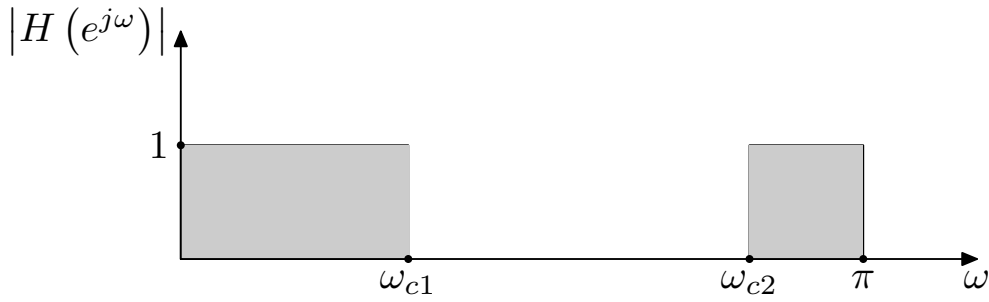


Figura 2.4: Filtro rejeita-faixa ideal

onde  $\Theta(\omega)$  denota a fase da resposta em frequência  $\angle D(e^{j\omega})$  e o parâmetro  $M$  é equivalente ao atraso desejado do filtro.

Conforme será visto na seção 2.3, em determinadas aplicações especializadas, necessita-se atender conjuntamente ambas especificações de magnitude e fase da resposta em frequência, necessitando, dessa forma, de técnicas de projeto distintas dos projetos mais convencionais de filtros. Adicionalmente, a técnica desenvolvida na seção 2.3 define a especificação da fase da resposta em frequência pelo seu atraso de grupo  $\tau(\omega)$ :

$$\tau(\omega) = -\frac{d\Theta(\omega)}{d\omega} \quad (2.6)$$

Dessa forma, o atraso de grupo em (2.6) é definido como o oposto da derivada da fase da resposta em frequência  $\Theta(\omega)$ . Esse atraso de grupo  $\tau(\omega)$  fornece o atraso, em amostras, introduzido pelo sistema numa senóide de frequência  $\omega$  [4].

De forma genérica, o projeto de filtros digitais, com as especificações dadas no domínio da frequência, envolve os seguintes estágios:

1. Especificação das propriedades desejadas (resposta em frequência) do sistema.
2. Aproximação das especificações usando um sistema discreto no tempo e causal.
3. Se o sistema aproximado atende as especificações, ir para o passo 4, caso contrário, voltar para o passo 1, adequando as especificações desejadas.
4. Realização do sistema.

Esse trabalho é focado apenas na aproximação das especificações.

Especificamente para filtros FIR, o sinal de saída  $y(n)$  desses filtros pode ser expresso pela seguinte relação:

$$y(n) = \sum_{i=0}^M h(i) x(n-i), \quad (2.7)$$

onde  $h(n)$  é a resposta ao impulso do filtro,  $x(n)$  é o sinal de entrada e  $M$  é a ordem do filtro.

Aplicando-se a transformada  $z$  na equação (2.7), chega-se a seguinte relação [4]:

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{i=0}^M b_i z^{-i}, \quad (2.8)$$

onde no caso de filtros FIR,  $b_i$ , os coeficientes do filtro, relacionam-se diretamente com a resposta ao impulso do filtro  $h(n)$ , ou seja,  $b_i = h(n)$ .

Na prática, a equação (2.8) pode ser realizada de várias formas distintas, usando como blocos básicos atrasos, multiplicadores e somadores [4].

O problema básico de projeto de filtros digitais FIR consiste então em aproximar as características da resposta em frequência ideal com um sistema que possua resposta em frequência  $H(e^{j\omega})$ , pela seleção apropriada dos coeficientes  $b_i$  em (2.8) [2].

O restante desse capítulo é organizado da seguinte forma. A seção 2.2 apresenta rapidamente o projeto de filtros FIR com fase linear, sendo que essa subcategoria de filtros não é exatamente o foco desse trabalho. A seção 2.3 apresenta de forma detalhada a postulação do problema de aproximação para o projeto de filtros FIR de fase não-linear, algumas aplicações desses filtros e fórmulas fechadas para a função de erro de aproximação do filtro, as quais serão abordadas no contexto da teoria de otimização no Capítulo 3.

## 2.2 PROJETO DE FILTROS FIR COM FASE LINEAR

### 2.2.1 Forma dos coeficientes

As técnicas de projeto de filtros digitais FIR dessa seção foram desenvolvidas particularmente para filtros com fase linear. Conforme será desenvolvido, essa restrição impõe simetria ou anti-simetria nos coeficientes  $h(n)$ .

Os filtros de fase linear conforme a equação (2.5), se caracterizam por apresentarem atraso de grupo  $\tau$  constante, e portanto sua resposta na frequência deve ser da seguinte forma [4]:

$$H(e^{j\omega}) = B(\omega) e^{-j\omega\tau + j\phi}, \quad (2.9)$$

onde  $B(\omega)$  é real e  $\tau$  e  $\phi$  são constantes. Assim, a resposta ao impulso  $h(n)$  de filtros com fase linear satisfaz [4]:

$$\begin{aligned}
h(n) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\omega}) e^{j\omega n} d\omega \\
&= \frac{1}{2\pi} \int_{-\pi}^{\pi} B(\omega) e^{-j\omega\tau + j\phi} e^{j\omega n} d\omega \\
&= \frac{e^{j\phi}}{2\pi} \int_{-\pi}^{\pi} B(\omega) e^{j\omega(n-\tau)} d\omega \\
&= e^{j\phi} b(n - \tau),
\end{aligned} \tag{2.10}$$

onde  $b(n)$  é a transformada de Fourier inversa em tempo discreto de  $B(\omega)$ .

Como  $B(\omega)$  é real,  $b(n) = b^*(-n)$  e da equação (2.10), tem-se que  $b(n) = e^{-j\phi} h(n + \tau)$ .

Logo, a partir das deduções acima, a resposta ao impulso do filtro tem que satisfazer:

$$e^{-j\phi} h(n + \tau) = e^{j\phi} h^*(-n + \tau) \Rightarrow h(n + \tau) = e^{2j\phi} h^*(-n + \tau) \tag{2.11}$$

Após as restrições de causalidade e duração finita dos coeficientes  $h(n)$  em (2.11) e fazendo os coeficientes do filtro serem reais, em [4] mostra-se que os coeficientes do filtro obedecem:

$$h(n) = (-1)^k h(M - n), \quad k \in Z \tag{2.12}$$

Nota-se que em (2.12) os coeficientes do filtro  $h(n)$  têm que ser simétricos ou anti-simétricos.

Por fim, com relação à resposta em frequência dos filtros FIR com fase linear, considerando coeficientes reais, tem-se:

$$H(e^{j\omega}) = B(\omega) e^{-j\omega \frac{M}{2} + j \frac{k\pi}{2}} \tag{2.13}$$

Para efeitos práticos, consideram-se apenas os casos  $k = 0, 1$  [4] e os tipos de filtros FIR de fase linear são classificados em:

- Tipo I:  $k = 0$  e  $M$  par.
- Tipo II:  $k = 0$  e  $M$  ímpar.
- Tipo III:  $k = 1$  e  $M$  par.
- Tipo IV:  $k = 1$  e  $M$  ímpar.

## 2.2.2 Projeto por amostragem na frequência

O método de projeto de filtros FIR por amostragem na frequência parte da idéia de se obterem coeficientes  $h(n)$  do filtro cuja DFT, transformada discreta de Fourier, corresponda exatamente

às amostras da resposta na frequência desejada  $D(\omega)$ . Para isso, convém notar que a DFT de uma sequência  $h(n)$  de comprimento  $N$  corresponde a amostras de sua transformada de Fourier nas frequências  $\omega = \frac{2\pi k}{N}$ , ou seja [4]:

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n) e^{-j\omega n} \quad (2.14)$$

e a partir de (2.14), com  $\omega = \frac{2\pi k}{N}$  obtém-se:

$$H\left(e^{j\frac{2\pi k}{N}}\right) = \sum_{n=0}^{N-1} h(n) e^{-j\frac{2\pi kn}{N}}, \text{ para } k = 0, \dots, N-1 \quad (2.15)$$

Dessa forma, no projeto de filtros FIR por amostragem na frequência,  $h(n)$  pode ser determinada amostrando-se a resposta na frequência desejada nos  $N$  pontos  $e^{j\frac{2\pi k}{N}}$  e encontrando-se sua DFT inversa. Ou seja, para a resposta desejada  $D(\omega)$ , é preciso primeiro achar [4]:

$$A(k) e^{j\theta(k)} = D\left(\frac{\omega_s k}{N}\right), \text{ para } k = 0, \dots, N-1, \quad (2.16)$$

onde  $A(k)$  e  $\theta(k)$  são as amostras das respostas desejadas de amplitude e fase, respectivamente. Para os filtros FIR de fase linear desenvolvidos no item 2.2.1, em [4] são desenvolvidas fórmulas fechadas para cada tipo de filtro, Tipo I, II, III e IV.

Embora o método de projeto de filtros FIR por amostragem na frequência seja bastante simples conceitualmente e de fácil aplicação, a principal razão pela qual este método não é muito utilizado deve-se a quantidade considerável de ondulação tanto nas bandas de passagem como nas bandas de rejeição do filtro resultante. Isso se deve, em parte, pela razão das fórmulas fechadas derivadas a partir de (2.16) garantirem que a transformada de Fourier de  $h(n)$  e a resposta desejada  $D(\omega)$  coincidam apenas nas  $M+1$  frequências distintas  $\frac{2\pi k}{M+1}$ ,  $k = 0, \dots, M+1$ , onde  $M$  é a ordem do filtro. Nas outras frequências não há qualquer controle sobre as ondulações resultantes.

### 2.2.3 Projeto por aproximação com funções-janela

Filtros ideais podem ser expressos por uma função transferência com número infinito de coeficientes  $h(n)$  conforme:

$$H(z) = \sum_{n=-\infty}^{\infty} h(n) z^{-n} \quad (2.17)$$

A relação entre a resposta em frequência  $H(e^{j\omega})$  e resposta ao impulso  $h(n)$  do filtro ideal é dada pelo seguinte par de equações:



$$\begin{aligned}
H(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} h(n) e^{-j\omega n} \\
h(n) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\omega}) e^{j\omega n} d\omega
\end{aligned} \tag{2.18}$$

A idéia principal no projeto de filtros FIR com funções janelas é a de truncar os coeficientes  $h(n)$  do filtro ideal em (2.18), definindo para esse fim, uma sequência genérica auxiliar  $h'(n)$  de comprimento finito que resulte num filtro de ordem  $M$  da forma [4]:

$$h'(n) = \begin{cases} h(n), & \text{para } |n| \leq \frac{M}{2} \\ 0, & \text{para } |n| > \frac{M}{2} \end{cases}, \tag{2.19}$$

onde  $M$  é par. A função de transferência resultante fica [4]:

$$H'(z) = h(0) + \sum_{n=1}^{\frac{M}{2}} (h(-n) z^n + h(n) z^{-n}) \tag{2.20}$$

A função de transferência (2.20) pode se tornar causal pela sua multiplicação por  $z^{-\frac{M}{2}}$  sem distorção da resposta de módulo do filtro além de manter sua resposta de fase linear. Entretanto, apenas pelo truncamento dos coeficientes do filtro em (2.19) gera-se um efeito indesejado.

Especificamente, devido à convergência lenta da série de Fourier  $h(n)$  quando aproxima funções que apresentam descontinuidades, geralmente o caso de respostas em frequência ideais, geram-se ondulações de elevada amplitude na resposta de módulo próximo às extremidades das descontinuidades. Essas ondulações são comumente chamadas de oscilações de Gibbs [4].

Logo, embora não se possa remover as ondulações de Gibbs, pode-se tentar controlar sua amplitude multiplicando a resposta ao impulso  $h(n)$  por uma função janela  $w(n)$ :

$$h'(n) = h(n) w(n) \tag{2.21}$$

Essa janela  $w(n)$  tem que ser projetada de forma a introduzir um mínimo de desvio em relação à resposta na frequência ideal [4].

No domínio da frequência, essa operação de multiplicação da janela pode ser vista como a convolução periódica entre as respostas na frequência ideal  $H(e^{j\omega})$  e da função janela  $W(e^{j\omega})$ :

$$H'(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\omega'}) W(e^{j(\omega-\omega')}) d\omega' \tag{2.22}$$

Logo, visando a produção da menor distorção possível na equação (2.22), uma série de funções janela  $w(n)$  foram desenvolvidas, entre elas podendo-se citar:

- Janela Retangular
- Janelas Triangulares
- Janelas de Hamming
- Janelas de Hanning
- Janela de Blackman
- Janela de Kaiser
- Janela de Dolph-Chebyshev

Uma função janela útil possui uma resposta em frequência (magnitude) que tenha um lobo principal estreito, uma altura relativa reduzida dos lobos secundários e uma rápida taxa de decaimento da altura dos lobos [3]. De forma geral, a largura do lobo principal relaciona-se com a largura da banda de transição de  $|H(e^{j\omega})|$ , enquanto que da altura relativa dos lobos secundários relaciona-se com o tamanho das ondulações de  $|H(e^{j\omega})|$ . Esses fatores não podem ser melhorados simultaneamente ao mesmo tempo [3]. Dessa forma, existe um compromisso entre a largura do lobo principal e a altura relativa dos lobos secundários. Algumas funções janela, como a janela de Kaiser, provêm um parâmetro que pode ser variado para controlar esse compromisso [3].

Embora o método de projeto com função janela apresente fórmulas fechadas e seja bastante simples, o mesmo ainda possui algumas limitações. Geralmente, deseja-se projetar um filtro que seja o melhor dado uma ordem de filtro  $M$  arbitrária. Não existe sentido em discutir essa situação na falta de um critério de aproximação [1]. Para o caso da janela retangular da equação (2.19), em [1], mostra-se que a mesma apresenta a melhor aproximação quadrática média para uma resposta em frequência desejada e uma ordem de filtro  $M$ . Entretanto, o método de janela não permite controle individual sobre os erros de aproximação em diferentes bandas. Tais desvantagens são compensadas com métodos que utilizam algoritmos de aproximação conforme será visto em 2.2.4.

## 2.2.4 Projeto de aproximação por otimização

Conforme vistos nos itens 2.2.2 e 2.2.3, embora as técnicas descritas possuam fórmulas fechadas para o projeto de filtros FIR de fase linear, não existe nenhum controle independente nesses métodos das ondulações nas faixas de passagem e rejeição. Tal controle independente da ondulação, além da garantia de uma solução ótima no sentido da aproximação da resposta em frequência desejada, só é possível através de algoritmos numéricos.

Para o projeto de filtros FIR de fase linear, considerando os quatro tipos de filtro abordados no item 2.2.1, deve-se expressar a resposta em frequência desses tipos de forma unificada. Em [4], mostra-se que a resposta em frequência  $H(e^{j\omega})$  de filtros FIR de fase linear pode ser expressa por:

$$H(e^{j\omega}) = e^{-j(\alpha\omega - \beta)} Q(\omega) P(\omega) = e^{-j(\alpha\omega - \beta)} A(\omega), \quad (2.23)$$

onde  $A(\omega) = Q(\omega) P(\omega)$ ,  $\alpha = \frac{M}{2}$ ,  $M$  é a ordem do filtro e para cada caso tem-se que:

- Tipo I:  $\beta = 0$  e  $Q(\omega) = 1$
- Tipo II:  $\beta = 0$  e  $Q(\omega) = \cos\left(\frac{\omega}{2}\right)$
- Tipo III:  $\beta = \frac{\pi}{2}$  e  $Q(\omega) = \sin(\omega)$
- Tipo IV:  $\beta = \frac{\pi}{2}$  e  $Q(\omega) = \sin\left(\frac{\omega}{2}\right)$

Como se deseja aproximar somente a resposta de módulo do filtro, seja  $D(\omega)$  a resposta de amplitude desejada. Define-se a função de erro ponderado de aproximação, ou função objetivo no contexto da teoria de otimização, pela seguinte relação:

$$E(\omega) = W_q(\omega) (D_q(\omega) - P(\omega)), \quad (2.24)$$

onde  $W_q(\omega) = W(\omega) Q(\omega)$ ,  $D_q(\omega) = \frac{D(\omega)}{Q(\omega)}$ ,  $W(\omega)$  é a função peso de otimização e  $P(\omega)$  é a resposta de módulo do filtro projetado.

Finalmente, para implementação em computadores digitais, a função de erro em (2.24) é discretizada e expressa em forma matricial conforme a equação (2.25) [4]:

$$\mathbf{e} = \mathbf{W}_q (\mathbf{d}_q \mathbf{U} \mathbf{p}) \quad (2.25)$$

com as seguintes definições:

$$\mathbf{e} = [E(\omega_1) \ E(\omega_2) \ \dots \ E(\omega_I)]^T \quad (2.26)$$

$$\mathbf{W}_q = \text{diag} [W_q(\omega_1) \ W_q(\omega_2) \ \dots \ W_q(\omega_I)] \quad (2.27)$$

$$\mathbf{d}_q = [D_q(\omega_1) \ D_q(\omega_2) \ \dots \ D_q(\omega_I)]^T \quad (2.28)$$

$$\mathbf{U} = \begin{bmatrix} 1 & \cos(\omega_1) & \cos(2\omega_1) & \dots & \cos(L\omega_1) \\ 1 & \cos(\omega_2) & \cos(2\omega_2) & \dots & \cos(L\omega_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \cos(\omega_I) & \cos(2\omega_I) & \dots & \cos(L\omega_I) \end{bmatrix} \quad (2.29)$$

$$\mathbf{p} = [p(0) \ p(1) \ \dots \ p(L)]^T, \quad (2.30)$$

onde  $I = KM$ , em casos práticos  $8 \leq K \leq 16$ ,  $\mathbf{p}$  são os coeficientes do filtro projetado e  $L$  é definido para cada tipo de filtro:

- Tipo I:  $L = \frac{M}{2}$
- Tipo II:  $L = \frac{M-1}{2}$
- Tipo III:  $L = \frac{M}{2} - 1$
- Tipo IV:  $L = \frac{M-1}{2}$

#### 2.2.4.1 Método dos mínimos quadrados

No método dos mínimos quadrados, a idéia é minimizar o quadrado da energia da função erro em (2.24) em função dos coeficientes do filtro  $\mathbf{p}$ :

$$\min_{\mathbf{p}} \{L_2(\mathbf{p})\} = \min_{\mathbf{p}} \left\{ \int_0^\pi |E(\omega)|^2 d\omega \right\} \quad (2.31)$$

Na função de minimização na equação (2.31), define-se  $L_2(\mathbf{p})$ . Essa quantia é equivalente à função de distância de aproximação, denominada norma  $L_p$ , com  $p = 2$ , conforme será visto no Capítulo 3. Ainda nesse capítulo, mostra-se que (2.31) é convexa globalmente, e possui um minimizador único global. Essa solução global, ou os coeficientes do filtro no sentido ótimo  $\mathbf{p}^*$ , são dados por [4]:

$$\mathbf{p}^* = (\mathbf{U}^T \mathbf{W}_q^2 \mathbf{U})^{-1} \mathbf{U}^T \mathbf{W}_q^2 \mathbf{d}_q \quad (2.32)$$

#### 2.2.4.2 Método de Chebyshev

No método de Chebyshev a idéia é minimizar o máximo valor absoluto da função de erro em (2.24) em função dos coeficientes do filtro  $\mathbf{p}$ :

$$\min_{\mathbf{p}} \{L_\infty(\mathbf{p})\} = \min_{\mathbf{p}} \left\{ \max_{\omega \in \Omega} \{|E(\omega)|\} \right\}, \quad (2.33)$$

onde  $\Omega$  é o conjunto de faixas de frequências utilizadas na otimização.

Novamente, na função de minimização na equação (2.33), define-se  $L_\infty(\mathbf{p})$ . Essa expressão é equivalente à norma  $L_p$ , com  $p = \infty$ , conforme será visto no Capítulo 3. Ainda nesse capítulo, mostra-se que (2.33) é convexa globalmente, e possui um minimizador único global. Entretanto, diferentemente do método dos mínimos quadrados, a solução não possui fórmula fechada. Para resolução desse problema utiliza-se o conhecido algoritmo iterativo de Parks-McClellan [5].

#### 2.2.4.3 Método WLS-Chebyshev

Para algumas aplicações especializadas, necessita-se que tanto a atenuação mínima (critério Chebyshev), quanto a energia total (critério dos mínimos quadrados) na faixa de rejeição sejam

considerados simultaneamente. Nesses casos, as soluções Chebyshev e dos mínimos quadrados não podem ser abordadas separadamente. Deve-se obter um procedimento de projeto com boas características com relação tanto à atenuação mínima quanto à energia total na faixa de rejeição [4].

Em [6] é definido o algoritmo WLS-Chebyshev, um algoritmo que aborda os dois critérios de otimização de forma conjunta. Sucintamente, esse algoritmo realiza uma atualização recursiva da matriz de pesos (módulo do erro na equação (2.25)), utilizando uma matriz envoltória que simplifica os cálculos e acelera a convergência da otimização.

## **2.3 PROJETO DE FILTROS FIR COM FASE NÃO-LINEAR - APROXIMAÇÕES ARBITRÁRIAS DE MÓDULO E FASE**

### **2.3.1 Introdução**

Para os métodos de amostragem na frequência seção 2.2.2, função janela seção 2.2.3 e nos métodos por otimização seção 2.2.4 de projeto de filtros digitais FIR, partiu-se do princípio que o filtro possuiria fase linear, conforme a restrição de simetria ou anti-simetria dos coeficientes  $h(n)$  do filtro na equação (2.12). Embora na maioria das situações de uso de filtros digitais deseje-se essa característica, em algumas aplicações especializadas essa característica pode ser mais restritiva do que desejado. Os seguintes cenários motivam a consideração de filtros FIR com fase não-linear [3]:

- Em algumas aplicações, linearidade de fase é necessária somente em determinadas bandas de frequência. Nesse caso, é então desejável sacrificar fase linear exata em troca de redução/controle do atraso do filtro.
- Em algumas aplicações é necessário que a resposta em frequência  $H(e^{j\omega})$  do filtro aproxime uma resposta em frequência não-linear desejada  $D(e^{j\omega})$ . Exemplos incluem projeto de equalizadores.

Neste trabalho, será utilizada a abordagem de aproximação arbitrária de magnitude e fase para o projeto de filtros FIR de fase não-linear com atraso reduzido nas bandas de passagem. Nessa situação, conforme citado nas considerações acima, a motivação para fase não-linear se deve à característica dos filtros FIR de fase linear possuírem atraso de grupo constante e independente da frequência em toda banda do filtro. Na prática, atraso de grupo variável nas bandas de rejeição é aceitável, e permitindo que a resposta de fase seja não-linear nas bandas de rejeição, os filtros FIR de fase não-linear podem ser projetados com atraso de grupo constante nas bandas de passagem. Dessa forma, esses filtros possuem atraso de grupo significativamente reduzidos em relação aos filtros FIR de fase linear projetados com atraso de grupo constante em todas as bandas de frequência do filtro [7].

Embora as características desejadas acima, fase linear e atraso de grupo reduzido nas bandas de passagem e fase não-linear nas bandas de rejeição, sejam prontamente satisfeitas com filtros IIR, algumas complexidades inerentes do projeto de filtros IIR com a abordagem da aproximação simultânea da magnitude e atraso de grupo do filtro são um possível empecilho na utilização de filtros IIR nessa situação em particular. Conforme amplamente difundido na literatura, os filtros IIR oferecem melhor seletividade e eficiência computacional, além de um atraso reduzido do sistema em comparação com um filtro FIR análogo [7]. Entretanto, as principais desvantagens do projeto de filtros IIR são que a resposta de fase linear pode ser alcançada apenas aproximadamente e que o projeto deve lidar com o problema de estabilidade, o qual não existe no caso dos filtros FIR [7]. Particularmente, o problema de estabilidade pode ser abordado de várias maneiras. Uma metodologia consiste na imposição de restrições de estabilidade, gerando um problema de otimização com restrição. Entretanto, dificuldades técnicas podem geralmente ocorrer quando as restrições de estabilidade sejam explicitamente relacionadas com as variáveis de projeto [7]. Quando as restrições nas variáveis são feitas de forma implícita, consegue-se apenas condições suficientes para estabilidade e que são em geral demasiadamente restritivas para permitir um projeto satisfatório [7].

O restante desse capítulo descreve detalhadamente o projeto de filtros FIR pela aproximação arbitrária de magnitude e fase. Primeiramente, em 2.3.2, o projeto de filtros FIR de fase não-linear é postulado como um problema clássico de aproximação. Essa formulação é de suma importância, já que possibilitará a aplicação da teoria de otimização ao problema proposto. Já no item 2.3.3, será definido a forma da função de transferência e fórmulas para a magnitude e atraso de grupo do filtro serão desenvolvidas. Por fim, no item 2.3.4 desenvolve-se o erro de aproximação de projeto do filtro sob a óptica da teoria de otimização.

### 2.3.2 Problema clássico de aproximação

Para filtros FIR de fase não-linear <sup>1</sup>, o projeto de seus coeficientes pode ser postulado como um problema clássico de aproximação [8]. Em termos gerais, o problema pode ser postulado como:

Seja  $f(x)$  uma função real dada definida num conjunto  $X$ , e seja  $F(A, x)$  uma função real de aproximação com dependência contínua em  $x \in X$  e em  $N$  parâmetros  $A$ . Dada uma função de distância  $\rho$ , determine os  $N$  parâmetros  $A^* \in P$  tal que:

$$\rho[F(A^*, x), f(x)] \leq \rho[F(A, x), f(x)], \quad (2.34)$$

para todo  $A \in P$ .

---

<sup>1</sup>A postulação do problema de aproximação para filtros IIR segue a mesma metodologia.

Para aplicar esse problema genérico de aproximação ao problema particular do projeto de filtros FIR de fase não-linear, considere a função de transferência  $H(z)$  do filtro como uma função de  $N$  parâmetros, ex: os zeros da função transferência, ordenados num vetor  $A$ . O conjunto  $P$  é definido em  $P \in \mathbb{C}$ . A variável independente  $x$  é convertida na frequência digital  $\omega$  e o conjunto  $X$  é definido em  $(\omega : 0 \leq \phi \leq \pi)$ . Adicionalmente, a resposta em frequência do filtro, incluindo a magnitude e atraso de grupo, podem ser expressas por uma função real de  $A$  e  $\omega$ ,  $F(A, \omega)$ , enquanto que a resposta em frequência desejada se torna  $f(\omega)$ .

Definido o problema de aproximação, um aspecto importante da otimização está relacionado à função de distância. A função utilizada nesse trabalho é a norma  $L_p$ , já que essa função é extensamente estudada e suas propriedades são bem conhecidas [8].

$$\|L(A)\|_p = \left( \int_0^\pi W(\omega) |F(A, \omega) - f(\omega)|^p d\omega \right)^{\frac{1}{p}} \quad (2.35)$$

Os critérios mais utilizados no projeto de filtros, tais como mínimos quadrados e Chebyshev ou minimax, são incluídos como casos especiais da equação (2.35).

### 2.3.3 Forma da função transferência e da resposta em frequência

A função de transferência do filtro FIR de fase não-linear é definida nesse trabalho na forma pólo-zero-ganho<sup>2</sup> conforme a equação (2.36).

$$H(z) = k_0 \prod_{i=1}^M z - z_{zi}, \quad (2.36)$$

onde  $z_{zi}$  são os zeros da função de transferência, na forma pólo-zero-ganho, reais ou em pares conjugados complexos,  $M$  é a ordem do filtro e  $k_0$  é a constante de ganho.

Expandindo o produtório na função de transferência em (2.36), obtém-se uma função de transferência na forma da equação (2.8), onde os coeficientes do filtro  $b_i$ , para filtros FIR de fase não-linear, não possuem mais a relação de simetria ou anti-simetria da equação (2.12).

Uma escolha adequada para o vetor de parâmetros  $A$  da função de transferência na forma pólo-zero-ganho da equação (2.36) é expressar os zeros da função de transferência em coordenadas polares conforme (2.37).

$$A = [\phi_{z1}, r_{zi}, \phi_{z2}, r_{z2}, \dots, \phi_{zM}, r_{zM}, a_N], \quad (2.37)$$

---

<sup>2</sup>No caso de filtros IIR, a função de transferência contém pólos e zeros. Sua forma fica então:

$$H(z) = k_0 \frac{\prod_{i=1}^n z - z_{zi}}{\prod_{i=1}^m z - z_{pi}}$$

onde o parâmetro  $a_N$  será definido no contexto da magnitude e fase da resposta em frequência, conforme as equações (2.45) e (2.46), o número total de parâmetros é  $N = M + 1$  e os zeros da função de transferência em coordenadas polares são expressos por:

$$z_{zi} = r_{zi} e^{j\phi_{zi}} \quad (2.38)$$

Embora na seção 2.3.2 nenhuma restrição tenha sido feita aos valores dos parâmetros em  $A$ , ou seja,  $P \in \mathbb{C}$ , três situações podem ser delineadas para os valores assumidos pelos zeros  $z_{zi}$  na função de transferência em (2.36).

Na primeira situação, uma classe importante de filtros pode ser gerada a partir da restrição de todos os zeros  $z_{zi}$  do filtro estarem dentro do círculo unitário:

$$r_{zi} \in [0, 1) \quad \text{para } i = 1, 2, \dots, M \quad (2.39)$$

Esses filtros são chamados de filtros de fase-mínima e apresentam a seguinte propriedade na fase de sua resposta em frequência [2]:

$$\angle \Theta(\pi) - \angle \Theta(0) = 0 \quad (2.40)$$

Outra classe de filtros é estabelecida a partir da restrição de todos os zeros  $z_{zi}$  do filtro estarem fora do círculo unitário:

$$r_{zi} > 1 \quad \text{para } i = 1, 2, \dots, M \quad (2.41)$$

Esses filtros são chamados de filtros de fase-máxima e apresentam a seguinte propriedade na fase de sua resposta em frequência [2]:

$$\angle \Theta(\pi) - \angle \Theta(0) = M\pi \quad (2.42)$$

Por fim, quando o filtro possui alguns zeros dentro do círculo unitário e o restante dos zeros fora do círculo unitário, o filtro é chamado de fase-mista.

Para os três casos, nenhuma restrição é feita com relação à fase do  $i$ -ésimo zero  $z_{zi}$ , ou seja:

$$\phi_{zi} \in [0, 2\pi) \quad (2.43)$$

Nesse trabalho serão abordados tanto filtros de fase-mínima quanto filtros de fase-mista.

Uma forma de representar a resposta em frequência do filtro é expressar a magnitude pela atenuação em decibéis e o atraso de grupo em amostras, conforme definida na equação (2.44).



$$\eta(\omega) = -20 \log_{10} |H(e^{j\omega})| \text{ dB} \quad (2.44)$$

$$\tau(\omega) = -\frac{d}{d\omega} (\angle H(e^{j\omega}))$$

Para a função transferência na forma pólo-zero-ganho (2.36), mostra-se em [9] que a atenuação em módulo é representada por:

$$\eta(A, \omega) = -\sum_{i=1}^M 10 \log_{10} (1 - 2r_{zi} \cos(\omega - \phi_{zi}) + r_{zi}^2) - a_N \quad (2.45)$$

E o atraso de grupo pela equação [9]:

$$\tau(A, \omega) = \sum_{i=1}^M \frac{1 - r_{zi} \cos(\omega - \phi_{zi})}{1 - 2r_{zi} \cos(\omega - \phi_{zi}) + r_{zi}^2} - a_N, \quad (2.46)$$

onde  $a_N = \log_{10} k_0$ , para a atenuação em módulo e  $a_N = \tau_0$ , para o atraso de grupo.

### 2.3.4 Função objetivo para otimização

Para a aproximação simultânea da magnitude e atraso de grupo do filtro FIR assimétrico, a norma  $L_p$  em (2.35), também denominada função de erro ou função objetivo no contexto da otimização, é simplificada para aproximação discreta conforme a equação (2.47). Esse passo é importante para a efetiva otimização da distância de aproximação em computadores digitais.

$$\begin{aligned} L_{2p,2q}^{\eta,\tau}(A) &= \delta \sum_{i=1}^I W_{\eta}(\omega_i) (\eta(A, \omega_i) - \eta_d(\omega_i))^{2p} \\ &+ (1 - \delta) \sum_{i=1}^I W_{\tau}(\omega_i) (\tau(A, \omega_i) - \tau_d(\omega_i))^{2q}, \end{aligned} \quad (2.47)$$

onde  $W_{\eta}$  é a função peso discreta da magnitude do filtro,  $\eta$  é a resposta de magnitude discreta do filtro projetado,  $\eta_d$  é a resposta de magnitude discreta desejada,  $W_{\tau}$  é a função peso discreta do atraso de grupo do filtro,  $\tau$  é o atraso de grupo discreto do filtro projetado e  $\tau_d$  é o atraso de grupo discreto desejado.

Convém ainda observar que o número de pontos discretos  $I$  na frequência  $\omega_i$  da equação (2.47), seguindo a mesma regra de discretização do projeto de filtros digitais de fase linear da seção 2.2.4, obedece a relação  $I = KM$ . Novamente, em casos práticos  $8 \leq K \leq 16$ .

Adicionalmente, como os métodos de otimização que serão apresentados no Capítulo 4 necessitam da informação do gradiente da função de erro que deve ser otimizada (2.47), definem-se então as derivadas parciais do vetor de parâmetros  $A$  em forma fechada em (2.48).

$$\begin{aligned}
\frac{\partial \eta}{\partial r_{zi}} &= 20 \log_{10}(e) \frac{-r_{zi} + \cos(\omega - \phi_{zi})}{1 - 2r_{zi} \cos(\omega - \phi_{zi}) + r_{zi}^2} \\
\frac{\partial \eta}{\partial \phi_{zi}} &= 20 \log_{10}(e) \frac{r_{zi} \sin(\omega - \phi_{zi})}{1 - 2r_{zi} \cos(\omega - \phi_{zi}) + r_{zi}^2} \\
\frac{\partial \tau}{\partial r_{zi}} &= \frac{-(1 + r_{zi}^2) \cos(\omega - \phi_{zi}) + 2r_{zi}}{(1 - 2r_{zi} \cos(\omega - \phi_{zi}) + r_{zi}^2)^2} \\
\frac{\partial \tau}{\partial \phi_{zi}} &= \frac{-r_{zi} (1 - r_{zi}^2) \sin(\omega - \phi_{zi})}{(1 - 2r_{zi} \cos(\omega - \phi_{zi}) + r_{zi}^2)^2}
\end{aligned} \tag{2.48}$$

Como as expressões acima das derivadas parciais do vetor de parâmetros  $A$  fazem parte das expressões das derivadas parciais da função de erro (2.47) em relação aos zeros da função transferência, obtém-se, por fim, as expressões das derivadas parciais de (2.47) que serão utilizadas pelo algoritmo de otimização, conforme (2.49).

$$\begin{aligned}
\frac{\partial L_{2p,2q}^{\eta,\tau}(\eta, \tau)(A, \tau_0)}{\partial r_{zi}} &= \delta \sum_{k=1}^I 2p W_{\eta}(\omega_k) \frac{\partial \eta}{\partial r_{zi}} (\eta(A, \omega_k) - \eta_d(\omega_k))^{2p-1} + \\
&\quad (1 - \delta) \sum_{k=1}^I 2q W_{\tau}(\omega_k) \frac{\partial \tau}{\partial r_{zi}} (\tau(A, \omega_k) - \tau_d(\omega_k))^{2q-1} \\
\frac{\partial L_{2p,2q}^{\eta,\tau}(\eta, \tau)(A, \tau_0)}{\partial \phi_{zi}} &= \delta \sum_{k=1}^I 2p W_{\eta}(\omega_k) \frac{\partial \eta}{\partial \phi_{zi}} (\eta(A, \omega_k) - \eta_d(\omega_k))^{2p-1} + \\
&\quad (1 - \delta) \sum_{k=1}^I 2q W_{\tau}(\omega_k) \frac{\partial \tau}{\partial \phi_{zi}} (\tau(A, \omega_k) - \tau_d(\omega_k))^{2q-1}
\end{aligned} \tag{2.49}$$

Conforme será visto no Capítulo 3 a função objetivo (2.47) é uma função não-linear multimodal<sup>3</sup>, diferentemente das funções (2.32) e (2.33) para otimização do projeto de filtros FIR de fase linear que são convexas e possuem um mínimo global único. A função objetivo (2.47) também é classificada como uma função multi-objetivo, já que objetivos conflitantes na sua resolução, resposta de magnitude e atraso de grupo com diferentes critérios, ex: minimax para magnitude e mínimos quadrados para atraso de grupo, coexistem simultaneamente durante a otimização.

---

<sup>3</sup>A mesma classificação se aplica para a função objetivo no projeto de filtros IIR. Entretanto, diferentemente da equação (2.47), a função objetivo no projeto de filtros IIR possui pólos da função transferência, e para sua resolução, deve-se impor a restrição dos zeros da função transferência estarem dentro do círculo unitário. Essa restrição garante o projeto de filtros IIR estáveis.



# 3 TEORIA DE OTIMIZAÇÃO APLICADA AO PROJETO DE FILTROS DIGITAIS

## 3.1 INTRODUÇÃO

Otimização pode ser definida como a ciência para obter as melhores soluções de determinados problemas matematicamente bem definidos, que são geralmente modelos da realidade física. Dessa forma, a otimização envolve o estudo de critérios ótimos para os problemas, a determinação de métodos algorítmicos para sua solução, o estudo da estrutura de tais métodos e da experimentação computacional em problemas reais [10]. Tais problemas envolvem analistas em pesquisa operacional, engenheiros e planejadores em problemas como projeto ótimo, alocação de recursos escassos, operações de planejamento industrial ou cálculos para achar a melhor trajetória de um míssil [11].

Uma medida óbvia de complexidade nesses problemas de otimização relaciona-se com seu tamanho em termos do número de variáveis ou número de restrições. Basicamente, em problemas reais, distinguem-se três classes [12]: problemas de pequena-escala com cinco ou menos variáveis e restrições, média-escala com mais de cinco e menos de cem variáveis e restrições, e por fim, larga-escala com mais de cem e talvez milhares de variáveis e restrições. Embora o limite do número de variáveis em cada classe acima possa ser mais flexível, diferenças substanciais nos métodos de resolução em cada classe de problema são observadas. Como regra geral [12], problemas de pequena e média escala podem ser resolvidos com métodos genéricos de otimização, os quais são objetivo de estudo desse trabalho e serão apresentados nesse capítulo. Entretanto, métodos de larga-escala necessitam de código sofisticado que explore estruturas especiais do problema [12].

Outra medida de complexidade relaciona-se com a função matemática que expressa o problema de otimização e suas restrições nos valores assumidos pelas variáveis. Basicamente, duas classes distinguem-se nesse quesito: otimização linear e otimização não-linear.

Primeiramente, na otimização linear, ambas funções objetivo e de restrição (ver seção 3.2) são funções lineares da variável do problema. Uma função linear  $f$  é definida para todo número real  $x$  por uma equação da seguinte forma [13]:

$$f(x) = ax + b, \quad (3.1)$$

onde  $b$  é a interceptação da reta com o eixo  $y$  e  $a$  é a inclinação da reta. A função em (3.1) é também chamada linear devido ao seu gráfico ser uma linha reta. A Figura 3.1 apresenta o exemplo da função linear  $f(x) = 2x + 1$ .

Quando a função objetivo ou pelo menos uma das funções de restrição forem funções não-

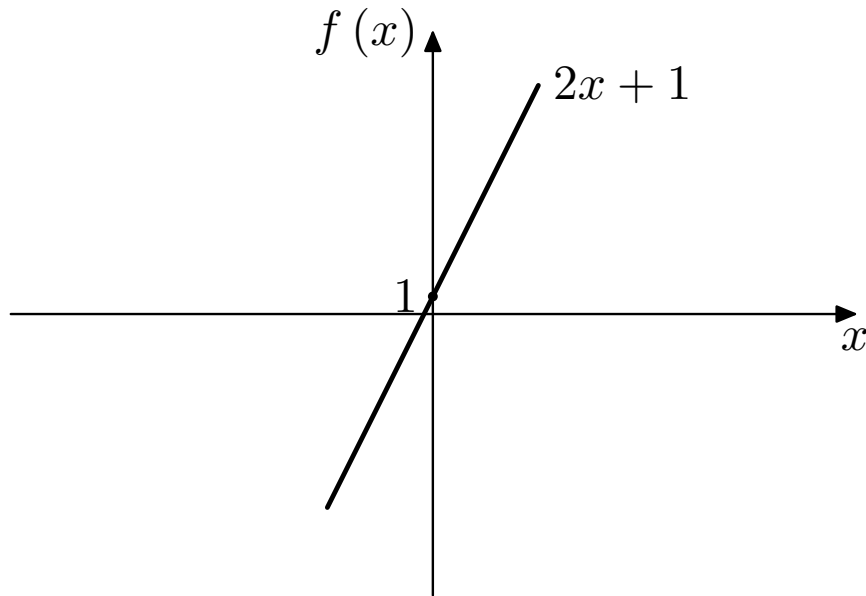


Figura 3.1: Exemplo de função linear

lineares da variável do problema, classifica-se o mesmo por otimização não-linear. Nessa classe de problemas, um conceito importante é o de convexidade. O termo convexo pode ser aplicado tanto para conjuntos como para funções [14]. Um conjunto  $S \in \mathbb{R}^n$  é um conjunto convexo se o segmento em linha reta conectando dois pontos quaisquer em  $S$  permanece completamente dentro de  $S$ . Ou seja, para dois pontos quaisquer  $x \in S$  e  $y \in S$ , tem-se [14]:

$$\alpha x + (1 - \alpha) y \in S \quad \text{para todo } \alpha \in [0, 1] \quad (3.2)$$

Da mesma forma, uma função  $f$  é convexa se seu domínio  $S$  é um conjunto convexo e para quaisquer dois pontos  $x$  e  $y$  em  $S$ , a seguinte propriedade é satisfeita [14]:

$$f(\alpha x + (1 - \alpha) y) \leq \alpha f(x) + (1 - \alpha) f(y) \quad \text{para todo } \alpha \in [0, 1] \quad (3.3)$$

O conceito de convexidade de uma função tem ligação estrita com a complexidade de resolução do problema de otimização. Um importante resultado da teoria de otimização é que se ambas função objetivo e funções de restrição são convexas, então qualquer solução local do problema é de fato uma solução global [14].

Qualquer transformação linear que possua valores em  $\mathbb{R}$  é convexa [15], logo, qualquer solução local de um problema de otimização linear é de fato a solução global. Além das funções lineares, várias outras famílias de funções também são convexas, como a família de funções quadráticas [15]:

$$f(x) = ax^2 + bx + c \quad \text{com } a \neq 0 \quad (3.4)$$

A Figura 3.2 apresenta o gráfico da função quadrática  $f(x) = x^2$ .

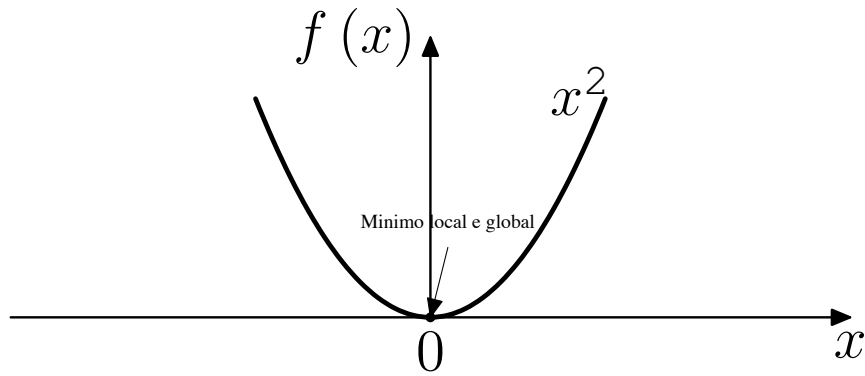


Figura 3.2: Exemplo de função quadrática

Entretanto, de forma geral, função não-lineares não apresentam a propriedade de convexidade da equação (3.3). Essa observação aumenta consideravelmente a complexidade da otimização, já que para funções não-lineares com vários mínimos locais é difícil achar um mínimo global, devido ao algoritmo de otimização apresentar tendência de aprisionamento nos mínimos locais [14]. A Figura 3.3 apresenta uma função não-linear com múltiplos mínimos locais.

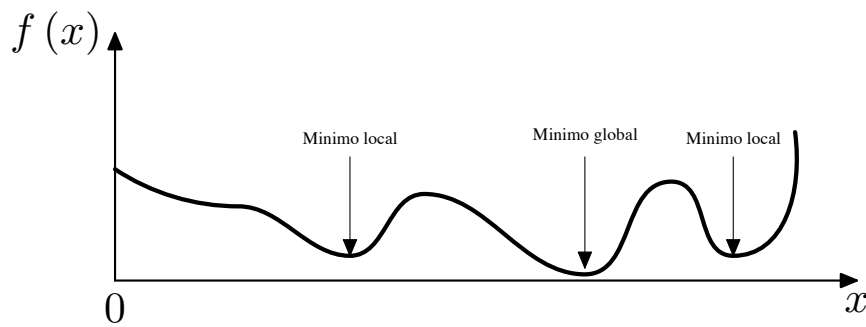


Figura 3.3: Exemplo de função não-linear

Convém observar que os algoritmos que serão apresentados no Capítulo 4 não apresentam matematicamente a garantia de soluções globais para funções objetivos com múltiplos mínimos locais. Entretanto, embora os algoritmos apresentados no Capítulo 5 também não apresentem matematicamente essa garantia, grande parte da literatura vem apresentando melhores resultados desses algoritmos na classe de funções não-lineares com múltiplos mínimos locais [16]. Os resultados que serão apresentados no Capítulo 7 também confirmam essa afirmação.

Ainda com relação aos algoritmos de otimização, sua operação de busca da solução do problema tipicamente envolve a seleção de um vetor inicial  $\mathbf{x}_0$ , e a partir desse vetor é gerado um vetor melhorado  $\mathbf{x}_1$ . Da mesma forma, uma sequência de pontos cada vez mais próximos da solução  $\mathbf{x}^*$  é gerada  $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k, \dots$ . Para problemas de otimização linear, a sequência gerada é de tamanho finito, alcançando a solução exatamente após um número finito de passos [12]. Para problemas de otimização não-linear, a sequência gerada nem sempre alcança a solução mas sim, converge para ela. Dessa forma, o processo nesses casos é terminado quando um ponto suficientemente próximo à solução, para propósitos práticos, é obtido [12].

O restante desse capítulo é organizado da seguinte forma. A seção 3.2 define matematicamente o problema de otimização, assim como apresenta alguns termos comuns em uso na literatura. A seção 3.3 classifica os problemas de otimização em certas classes que geralmente requerem alterações substanciais na forma de resolução do problema. Essas classes são problemas com e sem restrição e problemas mono-objetivo e multi-objetivos. A seção 3.4 apresenta análise detalhada do problema de otimização no projeto de filtros FIR de fase linear apresentados na seção 2.2 do Capítulo 2. Por fim, a seção 3.5 apresenta detalhadamente a análise do problema de otimização para filtros FIR de fase não-linear apresentados na seção 2.3 do Capítulo 2.

## 3.2 DEFINIÇÃO DO PROBLEMA DE OTIMIZAÇÃO

O problema de otimização pode ser definido nos seguintes termos [17]:

Dado um conjunto  $X$  e uma função  $f: X \rightarrow \mathbb{R}$ , também chamada de função objetivo, deseja-se achar um ponto  $\mathbf{x}^* \in X$  tal que, para todo  $\mathbf{x} \in X$ , mantém-se a relação  $f(\mathbf{x}) \geq f(\mathbf{x}^*)$ . A variável  $\mathbf{x}$  é geralmente chamada de variável de controle ou decisão. Dessa forma, o problema de otimização é dado por:

$$\begin{aligned} \min f(\mathbf{x}) \quad & \mathbf{x} \in \mathbb{R}^n \\ \text{sujeito a} \quad & \\ c_j(\mathbf{x}) \geq 0 \quad & j \in I \\ c_j(\mathbf{x}) = 0 \quad & j \in E \end{aligned} \quad (3.5)$$

onde no problema de otimização em (3.5), a variável de controle está contida <sup>1</sup> no espaço euclidiano  $n$ -dimensional  $\mathbf{x} \in \mathbb{R}^n$ ,  $c_j: \mathbb{R}^n \rightarrow \mathbb{R}$  denotam as funções de restrição que a variável de controle  $\mathbf{x}$  deve satisfazer para minimizar  $f(\mathbf{x})$  e  $I$  e  $E$  são dois conjuntos disjuntos de inteiros que representam os índices das funções de restrições de desigualdade e igualdade, respectivamente.

O ponto  $\mathbf{x}^*$  da solução do problema de otimização em (3.5) pode ser um minimizador global, local ou local estrito. Um minimizador global  $\mathbf{x}^*$  de  $f$  possui a seguinte definição formal [14]:

Um ponto  $\mathbf{x}^*$  é um minimizador global se:

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \text{ para todo } \mathbf{x}, \quad (3.6)$$

---

<sup>1</sup>Para o projeto de filtros FIR de fase não-linear, conforme definido na seção 2.3, a variável de controle representada pelos zeros  $z_i$  da função transferência é definida no conjunto dos números complexos  $x \in \mathbb{C}^n$ . Entretanto, nesse trabalho, as mesmas definições do caso real são válidas para o domínio complexo dos zeros  $z_i$  da função transferência.

onde  $\mathbf{x}$  está definido para todo o conjunto dos números reais  $\mathbb{R}^n$ . Um exemplo de minimizador global pode ser visto na Figura 3.3. Ainda, conforme explanado na seção 3.1, o minimizador global pode ser difícil de achar, devido ao conhecimento da função  $f$  ser, na maioria das vezes, somente local.

Um minimizador local de  $f$ , por sua vez, é um ponto que alcança o menor valor de  $f$  na sua vizinhança [14]:

Um ponto  $\mathbf{x}^*$  é um minimizador local se existe uma vizinhança  $v$  de  $\mathbf{x}^*$  tal que:

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \text{ para todo } \mathbf{x} \in v \quad (3.7)$$

Por fim, um minimizador local estrito de  $f$  possui a seguinte definição:

Um ponto  $\mathbf{x}^*$  é um minimizador local estrito se existe uma vizinhança  $v$  de  $\mathbf{x}^*$  tal que:

$$f(\mathbf{x}^*) < f(\mathbf{x}) \text{ para todo } \mathbf{x} \in v \text{ com } \mathbf{x} \neq \mathbf{x}^* \quad (3.8)$$

A maioria dos algoritmos de solução do problema de otimização em (3.5) são capazes de achar apenas um minimizador local. Novamente, um exemplo de minimizador local pode ser visto na Figura 3.3.

### 3.3 CLASSIFICAÇÃO DOS PROBLEMAS DE OTIMIZAÇÃO

A partir da definição do problema de otimização na seção 3.2, quatro classes importantes de problemas podem ser derivadas a partir do problema genérico em (3.5). As duas primeiras classes relacionam-se com as restrições impostas na variável de controle  $\mathbf{x}$  do problema de otimização. Dessa forma, pode-se classificar o problema em (3.5) por otimização com restrição e otimização sem restrição. Especificamente, problemas de otimização com restrição são muito mais complexos do que problemas sem restrição. Os algoritmos de resolução de tais problemas necessitam de teoria adicional como multiplicadores de Lagrange e condições de primeira ordem [18]. Entretanto, dependendo da forma funcional das restrições impostas na variável de controle  $\mathbf{x}$ , certas eliminações e transformações de variáveis podem converter um problema com restrição no correspondente problema sem restrição [18]. Algumas transformações de variáveis serão apresentadas na seção 3.3.2, especialmente aplicáveis numa classe específica (fase-mínima) de filtros FIR de fase não-linear, conforme desenvolvido na seção 3.5.

Outras duas classes importantes de problemas relacionam-se com o número de funções objetivos simultâneas durante a otimização. O problema de otimização representado na equação



(3.5) define apenas uma função objetivo  $f$ . Esse tipo de problema é classificado por otimização mono-objetivo. O projeto de filtros FIR de fase linear com critério dos mínimos quadrados e critério Chebyshev da seção 2.2.4 do Capítulo 2 são exemplos de problemas de otimização mono-objetivo, conforme será desenvolvido na seção 3.4.

Entretanto, problemas reais geralmente envolvem tomada de decisão, escolhas e busca por compromissos entre vários objetivos conflitantes. Deseja-se então que todos esses objetivos sejam o melhor possível, em outras palavras, ótimos [19]. Problemas desse tipo são classificados por otimização multi-objetivo. Diferentemente dos problemas de otimização mono-objetivo, uma solução ótima no sentido que minimize todas as funções objetivos simultaneamente não existe necessariamente em problemas desse tipo. A decisão final deve ser feita levando em conta o equilíbrio total dos múltiplos objetivos [20]. O projeto de filtros FIR de fase não-linear desenvolvido na seção 2.3 do Capítulo 2 segue a formulação multi-objetivo, conforme será apresentado na seção 3.5.

### 3.3.1 Otimização sem restrição

Problemas de otimização sem restrição são definidos a partir da equação (3.5) com  $I = E = \emptyset$ . Mesmo para alguns problemas com restrições naturais nas variáveis, é mais seguro desconsiderar tais restrições já que elas não afetam a solução e não interferem com os algoritmos [14]. Todos os algoritmos apresentados no Capítulo 4 e 5 são estritamente projetados para problemas de otimização sem restrição. Mesmo no caso do projeto de filtros FIR de fase não-linear que necessitam de restrição na variável de controle, transformações serão aplicadas na variável para conversão do problema de otimização com restrição no seu correspondente problema sem restrição, conforme será visto na seção 3.5.

### 3.3.2 Otimização com restrição

Problemas de otimização com restrição apresentam funções adicionais de restrição nos valores assumidos pela variável de decisão  $\mathbf{x}$ . Essas funções de restrição são representadas por restrições de igualdade  $c_j(\mathbf{x}) = 0$  com  $j \in E$  e restrições de desigualdade  $c_j(\mathbf{x}) \geq 0$  com  $j \in I$ , conforme definidas no problema de otimização da equação (3.5).

Para problemas de otimização com restrição, define-se a região (conjunto) viável  $S$  por todos os pontos  $\mathbf{x}$  que satisfazem as restrições [14]:

$$S = \{\mathbf{x} \mid c_j(\mathbf{x}) = 0, j \in E; c_j(\mathbf{x}) \geq 0, j \in I\} \quad (3.9)$$

Pela definição da região viável  $S$ , uma forma mais compacta de representação do problema de otimização em (3.5) fica:

$$\min_{\mathbf{x} \in S} f(\mathbf{x}) \quad \mathbf{x} \in \mathbb{R}^n \quad (3.10)$$

Os problemas de otimização com restrição dividem-se, basicamente, em duas classes principais, programação com restrições lineares e programação com restrições não-lineares [18].

Na programação com restrições lineares, cada restrição  $c_j$  é uma função linear com forma apresentada na equação (3.1). Problemas desse tipo podem ser resolvidos por uma combinação de métodos de eliminação/transformação e métodos de conjuntos ativos [18]. Com essas técnicas, algoritmos de otimização projetados para resolução de problemas sem restrição podem ser facilmente aplicados aos problemas de otimização com restrição análogos. Nesse trabalho será apresentado apenas o método de eliminação nos problemas de otimização com restrição, já que esse método é capaz de lidar com restrições simples na variável de controle, encontrando assim, aplicação no projeto de filtros FIR de fase-mínima, apresentados na seção 2.3 do Capítulo 2.

A programação com restrições não-lineares é a classe de problemas de maior dificuldade de resolução na otimização com restrição. Entre os métodos de resolução dessa classe de problemas encontram-se a eliminação indireta pela resolução numérica de um sistema de equações não-lineares, que embora seja possível, geralmente não é eficiente, além de apresentar outras dificuldades numéricas [18]. Outro método consiste em transformar o problema de programação com restrições não-lineares em um problema de otimização sem restrição análogo pelo uso de funções de penalidade, embora a eficiência da resolução e a garantia de propriedades de convergência global sejam diretamente dependentes da correta escolha de uma função de penalidade viável [18]. Por fim, ainda é possível modelar o problema original de forma que sua resolução seja facilitada. Isso é feito pelo método de linearização das funções de restrição, onde aproximações pela série de Taylor são feitas nas funções de restrição, permitindo que sub-problemas com restrição linear sejam resolvidos a cada iteração do algoritmo de resolução do problema [18]. Nesse trabalho não serão desenvolvidos métodos para resolução de problemas em programação com restrições não-lineares, já que esses métodos fogem do escopo de aplicação em problemas de otimização de projeto de filtros digitais.

### 3.3.2.1 Método de eliminação e transformação de variáveis

O método de eliminação e transformação consiste em realizar operações no problema de otimização para reduzi-lo à uma forma de mais fácil resolução. Esse método pode ser vantajoso e uma série de transformações e eliminações foram desenvolvidas na literatura. Entretanto, é importante salientar que esse método não é completamente livre de riscos e que a solução original e a transformada podem não corresponder exatamente ou o método pode não funcionar adequadamente no problema transformado [18].

Para funções de restrição de igualdade, a possibilidade mais simples consiste no uso de equações que eliminam algumas das variáveis do problema. Essa técnica de resolução é conhecida como método da eliminação. Se no problema de otimização em (3.5), existirem  $m$  equações da

forma  $c(\mathbf{x}) = 0$  que podem ser arranjadas diretamente para resultar em [18]:

$$\mathbf{x}_1 = \phi(\mathbf{x}_2), \quad (3.11)$$

com  $\mathbf{x}_1$  e  $\mathbf{x}_2$  sendo partições de  $\mathbf{x}$  em  $\mathbb{R}^m$  e  $\mathbb{R}^{n-m}$ , então a função objetivo original é substituída por [18]:

$$\psi(\mathbf{x}_2) = f(\phi(\mathbf{x}_2), \mathbf{x}_2), \quad (3.12)$$

e dessa forma,  $\psi(\mathbf{x}_2)$  é minimizada sobre  $\mathbf{x}_2$  sem nenhuma restrição.

Os métodos de eliminação não são diretamente aplicáveis para restrições de desigualdade. Entretanto, algumas transformações de variáveis foram desenvolvidas na literatura na tentativa de simplificar o problema de otimização. Entre elas, é oportuno citar as seguintes transformações com aplicação na área de projeto de filtros digitais:

- $x_j \geq 0$ : Para essas restrições de desigualdade na variável de controle  $x$ , pode-se eliminá-las através da seguinte transformação [18]:

$$x_j = y_j^2, \quad (3.13)$$

onde  $y_j$  é uma nova variável definida para substituir  $x_j$ . Dessa forma, para qualquer  $y_j \in (-\infty, \infty)$  segue que  $x_j \geq 0$ .

- $l_j \leq x_j \leq u_j$ : Para restrições desse tipo, onde a variável de controle deve ser maior ou igual ao limite inferior  $l_j$  e menor ou igual ao limite superior  $u_j$ , pode-se aplicar a seguinte transformação [18]:

$$x_j = l_j + (u_j - l_j) \sin^2 y_i, \quad (3.14)$$

onde novamente  $y_j$  é uma variável definida para substituir  $x_j$ .

A vantagem que essas transformações oferecem é a de ampliar a gama de problemas de restrição com desigualdade que podem ser resolvidas com algoritmos de otimização projetados para problemas de otimização sem restrição. Entretanto, deve-se notar que as transformações em (3.13) e (3.14) introduzem distorções e não-linearidades adicionais na função objetivo  $f$  que as vezes podem desfavoráveis na resolução do problema [18].

### 3.3.3 Otimização mono-objetivo

O problema de otimização mono-objetivo foi definido na equação (3.5). Nessa classe de problemas, uma única solução que minimiza a função objetivo é encontrada e considerada como solução final, diferentemente da otimização multi-objetivo onde várias soluções finais do problema são igualmente possíveis, conforme será visto na seção 3.3.4. Convém lembrar que problemas dessa classe são a base da maioria dos problemas de otimização encontrados na literatura. Mesmos nos problemas de otimização multi-objetivo, técnicas de escalarização são aplicadas nesses problemas para conversão no correspondente problema de otimização mono-objetivo.

### 3.3.4 Otimização multi-objetivo

O problema de otimização multi-objetivo pode ser definido por [19]:

$$\begin{aligned} \min \{ & f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x}) \} \\ \text{sujeito a} & \\ & \mathbf{x} \in S \end{aligned} \quad (3.15)$$

onde no problema em (3.15) tem-se  $k \geq 2$  funções objetivo com  $f_i: \mathbb{R}^n \rightarrow \mathbb{R} \quad i = 1, \dots, k$ . O vetor de funções objetivos é representado por  $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})]^T$  pertencendo a região viável  $S$ , a qual é um subconjunto do espaço da variável de decisão  $\mathbb{R}^n$ . Na definição do problema de otimização em (3.15), a forma das funções de restrição não estão definidas, apenas referenciadas de forma geral como pertencentes ao conjunto  $S$ .

Para o problema de otimização em (3.15), o sentido da palavra minimizar significa a minimização de todas as funções objetivos simultaneamente. Se não existe conflito entre as funções objetivos, então uma solução única pode ser encontrada onde todas as funções objetivos alcançam seus valores ótimos. Nessa situação, nenhum método especial de otimização se faz necessário [19]. Entretanto, na otimização multi-objetivo, geralmente não existe uma única solução que seja ótima com respeito a todas as funções objetivos. Isso significa que as funções objetivos são parcialmente conflitantes [19]. O conceito de várias soluções para o problema de otimização está diretamente relacionado com o conjunto Pareto-ótimo, conforme será desenvolvido na seção 3.3.4.1.

#### 3.3.4.1 Caracterização das soluções ótimas: conjunto Pareto-ótimo

Geralmente não é possível achar uma única solução para o problema (3.15) que seja ótima para todos os objetivos simultaneamente. Entretanto, alguns vetores de funções objetivos  $\mathbf{f}(\mathbf{x})$  podem ser selecionados para inspeção. Esses vetores são aqueles nos quais nenhum de seus componentes  $f_i(\mathbf{x})$  podem ser melhorados sem a deterioração de pelo menos um dos outros componentes. Essa definição é geralmente chamada Pareto-ótimo, termo desenvolvido pelo economista-engenheiro italiano-francês Vilfredo Pareto [19]. A definição formal do conceito Pareto-ótimo é [19]:

Um vetor de decisão  $\mathbf{x}^* \in S$  é Pareto-ótimo se não existir outro vetor de decisão  $\mathbf{x} \in S$  tal que:

$$\begin{cases} f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*) & \text{para todo } i = 1, \dots, k \\ f_j(\mathbf{x}) < f_j(\mathbf{x}^*) & \text{para pelo menos um índice } j \end{cases} \quad (3.16)$$

De forma análoga um vetor objetivo  $\mathbf{f}^* \in F$  é Pareto-ótimo se não existir outro vetor objetivo  $\mathbf{f} \in F$  tal que:

$$\begin{cases} f_i \leq f_i^* & \text{para todo } i = 1, \dots, k \\ f_j < f_j^* & \text{para pelo menos um índice } j \end{cases} \quad (3.17)$$

Na definição em (3.17),  $\mathbf{f}^*$  é Pareto-ótimo se o vetor de decisão  $\mathbf{x}^*$  correspondente também é Pareto-ótimo.

A ilustração de uma região viável  $S \in \mathbb{R}^3$  e sua imagem, a região  $F \in \mathbb{R}^2$  pode ser vista na Figura 3.4. De forma geral, no problema de otimização multi-objetivo em (3.15) existem várias soluções Pareto-ótimas (número infinito) [19]. Pode-se, então, referenciar essas soluções como o conjunto Pareto-ótimo. Na Figura 3.4, esse conjunto Pareto-ótimo pode ser visto pela linha cheia do gráfico do conjunto  $F$ . Deve-se ainda observar que o conjunto Pareto-ótimo obedece as mesmas classificações com relação a convexidade de conjuntos, conforme definido na equação (3.2).

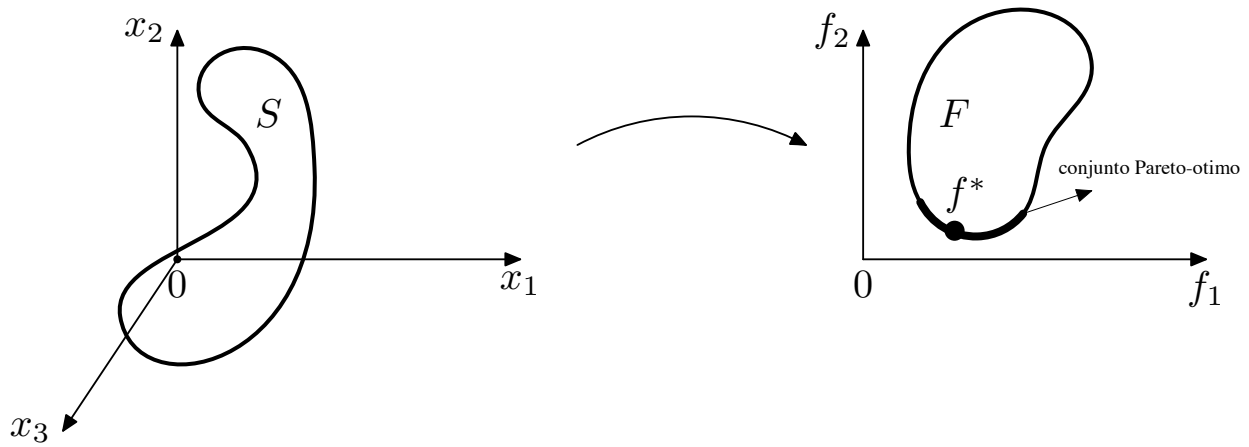


Figura 3.4: Exemplo dos conjuntos  $S$  e  $F$

Outro termo equivalente para o conjunto Pareto-ótimo é o de conjunto sem-dominância. Dessa forma, os vetores de decisão  $\mathbf{x}^*$  correspondentes às soluções incluídas no conjunto Pareto-ótimo são chamados de vetores sem-dominância [21].

O conceito Pareto-ótimo das equações (3.16) e (3.17) foi estabelecido no contexto global. Nesse contexto a palavra global possui o mesmo sentido de globalidade da definição de minimizador global na equação (3.6). Da mesma forma, encontrar um conjunto Pareto-ótimo global é uma tarefa difícil, especialmente se o conjunto Pareto-ótimo for um conjunto não-convexo. Faz-se então necessário definir o conceito Pareto-ótimo local [19]:

Um vetor de decisão  $\mathbf{x}^* \in S$  é Pareto-ótimo local se existe um  $\delta > 0$  tal que  $\mathbf{x}^*$  é Pareto-ótimo em  $S \cap B(\mathbf{x}^*, \delta)$

Um vetor objetivo  $\mathbf{f}^* \in F$  é Pareto-ótimo local se o vetor de decisão corresponde também é Pareto-ótimo local.

, nessa definição, a função  $B(\mathbf{x}^*, \delta)$  é uma bola aberta de raio  $\delta$  centrada no ponto  $\mathbf{x}^*$ .

Para finalizar, é interessante estabelecer uma relação entre a convexidade e globalidade das soluções Pareto-ótimo. Em [19], prova-se que se o problema de otimização é convexo, então cada solução Pareto-ótima local é também uma solução Pareto-ótima global.

### 3.3.4.2 Compromisso e tomada de decisão

De acordo com a definição Pareto-ótimo, mover-se de uma solução Pareto-ótima para outra envolve o compromisso entre os valores das funções objetivos  $f_i(\mathbf{x}^*)$  no vetor  $\mathbf{f}^*$ . Dessa forma, o compromisso reflete a razão de mudança nos valores das funções objetivos. Essa razão envolve o incremento de uma função objetivo, quando os valores de algumas outras funções objetivos decrescem. Uma definição importante nessa situação é a taxa de compromisso  $\Lambda_{ij}$  [22]:

$$\Lambda_{ij}(\mathbf{x}^1, \mathbf{x}^2) = \frac{f_i(\mathbf{x}^1) - f_i(\mathbf{x}^2)}{f_j(\mathbf{x}^1) - f_j(\mathbf{x}^2)}, \quad (3.18)$$

com  $f_j(\mathbf{x}^1) - f_j(\mathbf{x}^2) \neq 0$ , onde na definição em (3.18) tem-se  $i, j = 1, \dots, k, i \neq j, \mathbf{x}^1$  e  $\mathbf{x}^2$  são dois vetores decisão e  $\mathbf{f}(\mathbf{x}^1)$  e  $\mathbf{f}(\mathbf{x}^2)$  seus correspondentes vetores objetivos.

Outro conceito intimamente relacionado ao compromisso é o de tomada de decisão. Matematicamente, cada ponto Pareto-ótimo é uma solução igualmente aceitável para o problema de otimização multi-objetivo. Entretanto, geralmente deseja-se obter um único ponto como solução. Assim, a seleção de uma das soluções do conjunto Pareto-ótimo exige informação que não está contida nas funções objetivo [19].

Para essa situação, necessita-se da tomada de decisão para fazer a seleção de um ponto do conjunto Pareto-ótimo. Na otimização multi-objetivo, esse papel é exercido pelo tomador de decisão, o qual supõe-se possuir melhor conhecimento do problema e conseqüentemente, melhor discernimento de relações de preferência entre diferentes soluções. Geralmente, o tomador de decisão é responsável pela solução final [19].

Entretanto, pode ser difícil para o tomador de decisão distinguir entre soluções boas e ótimas em problemas reais. Se esse é o caso, a ênfase deve ser em achar boas soluções [19].

### 3.3.4.3 Escalarização do problema multi-objetivo

Vários métodos foram desenvolvidos na literatura para obtenção do conjunto Pareto-ótimo do problema de otimização multi-objetivo da equação (3.15). A estratégia mais comum consiste em

transformar o problema multi-objetivo em um problema análogo mono-objetivo através de sua escalarização.

Uma observação importante sobre os métodos de escalarização consiste no fato de que rotinas padrões para resolução de problemas de otimização mono-objetivo geralmente são capazes de garantir apenas solução local [19]. Por esse motivo, somente soluções Pareto-ótimas locais são geralmente obtidas quando funções de escalarização são utilizadas para resolução do problema multi-objetivo. Soluções Pareto-ótimas globais podem ser garantidas, por exemplo, se todas as funções objetivos e as regiões viáveis são convexas [19].

Os métodos de escalarização podem ser classificados de diversas formas. Uma possível forma de classificação é apresentada em [19], a qual leva em consideração a participação do tomador de decisão no processo de solução:

- Métodos sem-preferência: Nenhuma articulação de informação de preferência é utilizada pelo tomador de decisão.
- Métodos *a posteriori*: Articulação *a posteriori* de informação de preferência é utilizada pelo tomador de decisão.
- Métodos *a priori*: Articulação *a priori* de informação de preferência é utilizada pelo tomador de decisão.
- Métodos iterativos: Articulação progressiva de informação de preferência é utilizada pelo tomador de decisão.

Nesse trabalho serão abordados apenas métodos de escalarização *a posteriori*, devido principalmente à sua simplicidade e conveniência de implementação na área de projeto de filtros digitais.

Nos métodos *a posteriori*, também conhecidos por métodos para geração de soluções Pareto-ótimas [19], o conjunto Pareto-ótimo do problema de otimização multi-objetivo é encontrado e então apresentado ao tomador de decisão, o qual seleciona as soluções de maior preferência entre as alternativas apresentadas. Uma inconveniência desses métodos se deve ao fato de o processo de geração do conjunto Pareto-ótimo ser computacionalmente oneroso. Ainda mais, pode ser difícil para o tomador de decisão selecionar uma solução a partir de um grande conjunto de alternativas [19].

Entre as várias estratégias de escalarização nos métodos *a posteriori*, encontram-se [22]:

- Método ponderado: Faça  $W = \{w \mid w \in \mathbb{R}^n, w_j \geq 0 \text{ e } \sum_{j=1}^n w_j = 1\}$  ser um conjunto de pesos não-negativos. O método ponderado deve solucionar o seguinte problema de otimização mono-objetivo  $P(w)$  para algum vetor de pesos  $w \in W$ :

$$P(\mathbf{w}) = \min_{\mathbf{x} \in S} \sum_{j=1}^n w_j f_j(\mathbf{x}) \quad (3.19)$$

- Método do epsilon-restrito: O método do epsilon-restrito deve solucionar o seguinte problema de otimização mono-objetivo com restrição  $P_k(\epsilon)$ :

$$\begin{aligned} P_k(\epsilon) &= \min_{\mathbf{x} \in S} f_k(\mathbf{x}) \\ &\text{sujeito a} \\ f_j(\mathbf{x}) &\leq \epsilon_j \quad j = 1, \dots, n, j \neq k, \end{aligned} \quad (3.20)$$

onde  $\epsilon = (\epsilon_1, \dots, \epsilon_{k-1}, \dots, \epsilon_{k+1}, \dots, \epsilon_n)^T$ .

- Método da norma ponderada: O método da norma ponderada deve solucionar o seguinte problema de otimização mono-objetivo  $P_n(\mathbf{w})$ :

$$P_n(\mathbf{w}) = \min_{\mathbf{x} \in X} \sum_{j=1}^n w_j \left| f_j(\mathbf{x}) - \hat{f}_j \right|^p, \quad (3.21)$$

onde  $1 \leq p \leq \infty$  e  $\hat{f}_j = \min_{\mathbf{x} \in X} f_j(\mathbf{x})$ .

De forma geral, embora o método ponderado seja o mais simples dos métodos para geração de soluções Pareto-ótimas, o mesmo é falho em muitos casos de identificação completa de conjuntos Pareto-ótimos não-convexos [23].

O método epsilon-restrito seleciona uma função objetivo como objetivo primário para ser minimizado e converte todas outras funções objetivos em restrições pela introdução do vetor  $\epsilon$ . Embora o método epsilon-restrito seja aplicável em conjuntos Pareto-ótimos convexos e não-convexos, um possível empecilho nesse método consiste na sua complexidade, já que sistemas adicionais de restrição nos valores da função objetivo são impostos [23].

Mais especificamente com relação ao problema de conjuntos Pareto-ótimos no método ponderado, sem a pressuposição de convexidade, pela teoria da dualidade [22], algumas soluções do conjunto Pareto-ótimo nunca serão encontradas. Para problemas não-convexos, isso significa a não-existência de um hiperplano de suporte para a região viável no espaço das funções objetivos  $F$  [22]. Esse fenômeno é ilustrado no caso bidimensional da Figura 3.5. O ponto  $d$ , o qual coincide com a curva  $abcefg$  é não-convexo. Quaisquer pontos nas porções convexas da curva,  $abc$  e  $efg$  podem ser descobertos pelo método ponderado  $P(\mathbf{w})$  pela seleção apropriada de  $\mathbf{w}$ . Por exemplo, as soluções em  $b$  e  $f$  podem ser geradas pela resolução, respectivamente, de  $P(\mathbf{w}^1)$  e  $P(\mathbf{w}^2)$ . Por outro lado, nunca pode-se achar um vetor de pesos  $\mathbf{w}$  tal que a linha  $w_1 f_1 + w_2 f_2$  seja tangente ao ponto  $d$ . Nesse caso, diz-se que ocorreu uma interrupção por dualidade [22].



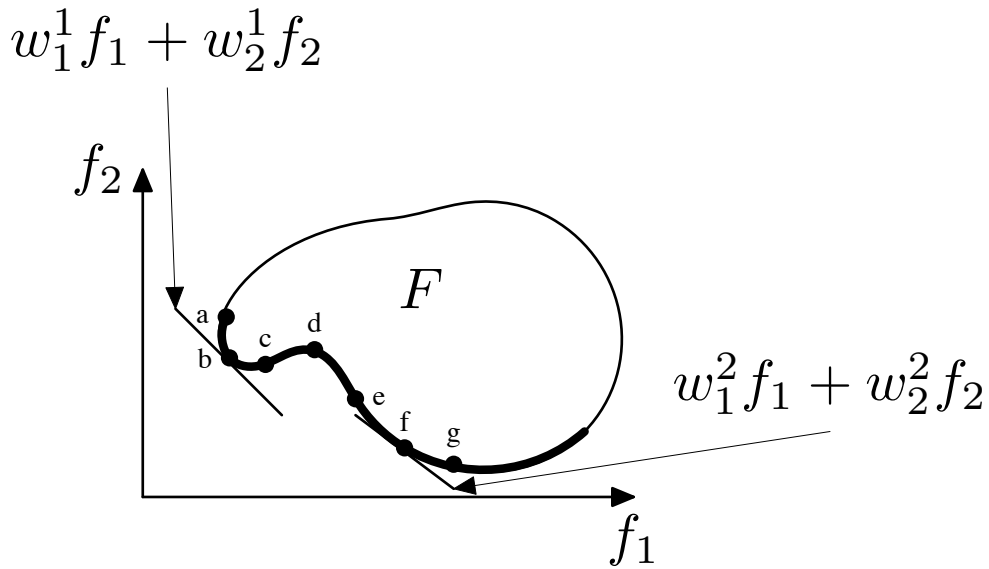


Figura 3.5: Dualidade em problemas não-convexos

Nesse trabalho será utilizado apenas o método ponderado para resolução do problema multi-objetivo da equação (3.15). Embora o autor tenha consciência de que esse método não é capaz de gerar o conjunto Pareto-ótimo por inteiro em casos não-convexos, o mesmo é capaz de gerar boas aproximações, além de ser eficiente e de simples implementação. Além disso, transformações foram desenvolvidas em [23], as quais podem ser aplicadas no problema de otimização multi-objetivo em (3.15) de forma que o conjunto Pareto-ótimo não-convexo se transforme em um conjunto Pareto-ótimo convexo análogo. Entretanto, algumas limitações são impostas nas transformações em [23] de forma que um estudo mais aprofundado deve ser feito para a correta aplicação dessas transformações no projeto de filtros digitais.

### 3.4 ANÁLISE DO PROBLEMA DE OTIMIZAÇÃO NO PROJETO DE FILTROS FIR DE FASE LINEAR

Conforme visto no Capítulo 2, um aspecto que pode ser tratado de forma bastante eficiente com otimização é o problema de aproximação no projeto de filtros digitais, onde os parâmetros do filtro devem ser escolhidos para alcançar uma resposta em frequência desejada.

O projeto de filtros FIR de fase linear apresentados na seção 2.2.4 do Capítulo 2 pode ser categorizado por problemas de otimização mono-objetivo sem restrição. O que se segue nessa seção é uma análise detalhada desses problemas, mínimos quadrados ou norma  $L_2$  e minimax ou norma  $L_\infty$ , sobre a óptica de otimização, levando em consideração a convexidade e complexidade de resolução do problema.

### 3.4.1 Filtros com critério dos mínimos quadrados

A análise subsequente do problema de otimização para o projeto de filtros FIR de fase linear com critério dos mínimos quadrados será feita apenas para filtros de fase linear do tipo I.

A resposta em frequência genérica de filtros FIR de fase linear apresentada na equação (2.23) pode ser simplificada para o tipo I, fazendo  $\beta = 0$  e  $Q(\omega) = 1$ . Assim:

$$H(e^{j\omega}) = e^{-j\omega \frac{M}{2}} P(\omega), \quad (3.22)$$

onde  $P(\omega)$  é expressa por [7]:

$$P(\omega) = \sum_{n=0}^{\frac{M}{2}} \mathbf{p}'(n) \cos \omega n, \quad (3.23)$$

onde os coeficientes do filtro FIR tipo I,  $\mathbf{p}'$ , são definidos por [7]:

$$\mathbf{p}'(n) = \begin{cases} h_{M/2} & \text{para } n = 0 \\ 2h_{M/2-n} & \text{para } n \neq 0 \end{cases} \quad (3.24)$$

Considerando a resposta em frequência desejada  $D(e^{j\omega})$  da forma:

$$D(e^{j\omega}) = e^{-j\omega \frac{M}{2}} P_d(\omega),$$

então, a função objetivo em (2.31) pode ser expressa pela distância de aproximação entre  $P_d(\omega)$  e  $P(\omega)$ , também chamada norma  $L_p$  com  $p = 2$ , conforme definida em (2.35):

$$f(\mathbf{p}') = L_2(\mathbf{p}') = \int_0^\pi W(\omega) [P(\omega) - P_d(\omega)]^2 d\omega \quad (3.25)$$

Considerando a função auxiliar  $\mathbf{c}(\omega)$ :

$$\mathbf{c}(\omega) = \left[ 1 \cos \omega \dots \cos \omega \frac{M}{2} \right]^T, \quad (3.26)$$

$P(\omega)$  pode ser escrito em termos do produto interno  $\mathbf{p}'^T \mathbf{c}(\omega)$  e a função objetivo  $f(\mathbf{p})$  em (3.25) pode ser expressa por uma função quadrática com respeito à  $\mathbf{p}'$ , na forma [7]:

$$f(\mathbf{p}') = \mathbf{p}'^T \mathbf{Q} \mathbf{p}' - 2\mathbf{p}'^T \mathbf{b} + k, \quad (3.27)$$

onde  $k$  é uma constante e  $\mathbf{Q}$  e  $\mathbf{b}$  são definidos por:

$$\mathbf{Q} = \int_0^\pi W(\omega) \mathbf{c}(\omega) \mathbf{c}^T(\omega) d\omega \quad (3.28)$$

$$\mathbf{b} = \int_0^\pi W(\omega) P_d(\omega) \mathbf{c}(\omega) d\omega, \quad (3.29)$$

a matriz  $\mathbf{Q}$  é positiva-definida e assim, a função objetivo  $f(\mathbf{p}')$  em (3.27) é convexa e possui um minimizador global único [7]. A solução fechada desse minimizador foi definida na equação (2.32).

### 3.4.2 Filtros com critério minimax ou Chebyshev

Considere que o conjunto de frequências  $\Omega$  na função de erro de projeto de filtros FIR de fase linear da equação (2.24) esteja contido em  $(\Omega: 0 \leq \omega \leq \pi)$ . A condição para a solução ótima da função objetivo em (2.33) pode ser postulada através da teoria da aproximação Chebyshev em conjuntos compactos [5]. Essa teoria garante que a função objetivo em (2.33) possui uma solução global única desde que as condições suficientes e necessárias que caracterizem a solução ótima obedeçam o seguinte teorema da alternância Chebyshev [5]:

Faça  $\Omega$  ser um subconjunto fechado qualquer de  $[0, \pi]$ . Em ordem que  $P(\omega)$ , a resposta em frequência do filtro projetado, seja a melhor aproximação global única de  $D(\omega)$ , a resposta em frequência desejada, em  $\Omega$ , é necessário e suficiente que a função de erro de projeto  $E(\omega)$  em (2.24) exiba em  $\Omega$  pelo menos  $L+2$  "alternações".

No teorema da alternância Chebyshev a quantia  $L$  foi definida na seção 2.2.4 em função dos filtros FIR de fase linear tipos I, II, III e IV. As alternâncias em frequência no teorema são definidas por [5]:

$$E(\omega_i) = -E(\omega_{i-1}) = \|E\|, \quad (3.30)$$

onde  $\omega_0 < \omega_1 < \dots < \omega_{L+1}$  e  $\omega_i \in \Omega$ . O operador  $\| \cdot \|$  na definição das alternâncias em (3.30) é definido por  $\|E\| = \max_{\omega \in \Omega} |E(\omega)|$ .

Embora a função objetivo em (2.33) possua um minimizador global único que obedece as condições do teorema da alternância Chebyshev, não existe uma fórmula fechada para obtenção desse minimizador. O método mais difundido de obtenção desse minimizador consiste no algoritmo iterativo de Parks-McClellan [5]. Esse algoritmo, por sua vez, é baseado no algoritmo de troca de Remez [24], o qual obtém iterativamente a melhor aproximação polinomial sob as condições do teorema da alternância.

### 3.5 ANÁLISE DO PROBLEMA DE OTIMIZAÇÃO NO PROJETO DE FILTROS FIR DE FASE NÃO-LINEAR

A formulação original do problema clássico de aproximação aplicado ao projeto de filtros digitais, seção 2.3.2, onde a norma  $L_p$  é utilizada como métrica da distância de aproximação entre as respostas desejadas de módulo e atraso de grupo do filtro projetado foi pioneiramente formulado em [8]. O presente trabalho segue essa formulação, e a função objetivo para projeto de filtros FIR de fase não-linear apresentada na equação (2.47) caracteriza-se como uma função não-linear multimodal [8]. Convém notar então que, devido à essa propriedade de não-linearidade, as soluções finais obtidas para a função objetivo em (2.47) não são necessariamente a solução global, conforme a definição de solução global da equação (3.6). O máximo que se pode afirmar sobre essas soluções é a propriedade local da equação (3.7).

Embora na formulação original do problema de aproximação simultânea de módulo e atraso de grupo em filtros digitais feitas em [8] não haja nenhuma menção sobre a natureza multi-objetivo da função objetivo em (2.47), claramente essa função possui forma semelhante à do método ponderado de escalarização da equação (3.19). Especificamente, nessa formulação multi-objetivo há duas funções distintas para minimização, a distância de aproximação da atenuação em decibéis do filtro:

$$f_1(\mathbf{x}) = \left( \int_0^\pi W_\tau(\omega) |\tau(A, \omega) - \tau_d(\omega)|^p d\omega \right)^{\frac{1}{p}}, \quad (3.31)$$

e a distância de aproximação do atraso de grupo:

$$f_2(\mathbf{x}) = \left( \int_0^\pi W_\eta(\omega) |\eta(A, \omega) - \eta_d(\omega)|^p d\omega \right)^{\frac{1}{p}}, \quad (3.32)$$

onde a variável de controle  $\mathbf{x}$  em (3.31) e (3.32) representa o vetor  $A$  de zeros da função de transferência em coordenadas polares.

Dessa forma, a formulação multi-objetivo do problema de aproximação arbitrária de módulo e atraso de grupo apresentada na seção 2.3 fica:

$$\begin{aligned} \min \{ & f_1(\mathbf{x}), f_2(\mathbf{x}) \} \\ \text{sujeito a} & \\ & \mathbf{x} \in S \end{aligned} \quad (3.33)$$

Duas classes distintas de problemas podem ocorrer com relação ao conjunto viável  $S$  em (3.33). Conforme discutido na seção 2.3, os filtros FIR de fase não-linear são classificados em filtros de fase mista ou filtros de fase mínima/máxima. No projeto de filtros FIR de fase mista, nenhuma restrição é feita nos valores assumidos pela variável de controle. Nesse caso,  $S = \emptyset$  e o problema é classificado como sem restrição. Entretanto, para filtros de classes mínima/máxima

restrições são impostas na variável de controle e dessa forma,  $S \neq \emptyset$ . Nesses casos, classifica-se o projeto de filtros de fase mínima/máxima por problemas de otimização com restrição.

Especificamente, nesse trabalho será abordado o projeto de filtros FIR de fase mínima. Nesse caso, conforme discutido na seção 2.3, os zeros da função transferência devem estar dentro do círculo unitário, conforme expresso na equação (2.39). Devido à simplicidade da função de restrição nesse problema de otimização, optou-se por convertê-lo em um problema de otimização sem restrição análogo, aplicando a transformação de variáveis da equação (3.14) na função objetivo (2.47).

O problema multi-objetivo em (3.33) pode ser resolvido pela sua escalarização com o método ponderado da equação (3.19). Especificamente, os pesos  $w_1$  e  $w_2$  do método estão representados na função objetivo em (2.47) por  $w_1 = \delta$  e  $w_2 = (1 - \delta)$ .

O problema da aproximação simultânea de módulo e atraso de grupo em filtros digitais com a formulação multi-objetivo foi tratado pioneiramente em [25], onde a formulação das funções objetivos (distâncias de aproximação) em (3.31) e (3.32) foram feitas em torno do critério Chebyshev ou minimax.

Uma consequência da formulação multi-objetivo em (3.33) consiste no fato de que não é possível obter uma única solução ótima para o problema, mas sim, uma família de soluções Pareto-ótimas. Dessa forma, o esforço computacional de resolução do problema multi-objetivo em (3.33) deve se concentrar em obter o conjunto de vetores decisão  $\mathbf{x}^*$  que obedeçam a relação em (3.16). Adicionalmente, deve se observar que devido à natureza não-linear das funções objetivos em (3.31) e (3.32), pode-se apenas garantir a obtenção de um conjunto Pareto-ótimo local. Infelizmente, ainda não existe teoria disponível para prever as características de filtros Pareto-ótimos globais em termos de padrões dos pólos ou comportamentos da resposta em frequência do filtro obtido [25]. Convém notar ainda que, em [25], a validação em termos da solução ótima global é feita apenas para os dois casos extremos do conjunto Pareto-ótimo, ou seja, para as configurações de pesos  $w_1 = 1, w_2 = 0$  e  $w_1 = 0, w_2 = 1$ . Como em [25] a caracterização das funções objetivos (3.31) e (3.32) é feita apenas em torno do critério Chebyshev, para as duas configurações de pesos é possível obter a solução ótima global. Todos os outros pontos restantes obtidos do conjunto Pareto-ótimo não possuem a garantia de solução ótima global.

Ainda com relação à obtenção do conjunto Pareto-ótimo, em [25], utiliza-se do método ponderado de escalarização do problema multi-objetivo em (3.33). Entretanto, em nenhuma parte do trabalho são feitas considerações sobre a possível não-convexidade do conjunto e a problemática de aplicação do método ponderado nessas situações, conforme discutido na seção 3.3.4.3 do presente trabalho.

A partir do conjunto Pareto-ótimo local obtido no problema multi-objetivo em (3.33), é tarefa do tomador de decisão, ou seja o projetista do filtro digital, escolher um ponto no conjunto Pareto-ótimo que atenda suas especificações. Essa escolha é feita pelo tomador de decisão já que o mesmo possui conhecimento e discernimento de relações de compromisso entre as diferentes

soluções do conjunto Pareto-ótimo obtido.

Com relação aos algoritmos utilizados para a solução do problema multi-objetivo em (3.32), várias técnicas foram apresentadas na literatura. O primeiro registro de resolução do problema multi-objetivo em (3.33) foi reportado em [8]. Nesse trabalho, utilizou-se do algoritmo Quasi-Newton para resolução do problema de projeto de filtros IIR com aproximação simultânea da resposta de módulo e atraso de grupo do filtro. O Capítulo 4 apresenta discussão detalhada sobre a família de algoritmos baseados em gradiente, da qual faz parte o algoritmo Quasi-Newton empregado em [8].

Em [25], o problema postulado na forma do método ponderado foi resolvido através do algoritmo de Programação Quadrática Sequencial (SQP). Embora esse algoritmo não seja abordado no presente trabalho, em [25] reporta-se baixa performance para filtros de alta ordem.

Já em [26], o mesmo método ponderado é resolvido com extensões *ad hoc* do algoritmo de correção diferencial. Com esse algoritmo reportou-se melhor performance, especialmente na obtenção do conjunto Pareto-ótimo de filtros FIR de fase não-linear.

Em [27], uma abordagem completamente distinta é empregada. Especificamente, utiliza-se da formulação multi-objetivo com algoritmos genéticos de seleção não-dominante elitista (ENSGA). Nesse algoritmo a natureza multi-objetivo do problema é fortemente ligada à atribuição da função de adequação e à estratégia de seleção [27].

Em [28] o problema multi-objetivo postulado na forma do método ponderado é resolvido com três algoritmos distintos da área de *soft-computing*: algoritmos genéticos, resfriamento simulado e otimização por enxame de partículas. Comparativos em termos da performance e qualidade das soluções obtidas foram feitos entre esses três algoritmos, além da comparação adicional com o algoritmo Quasi-Newton empregado em [8], e reportou-se o algoritmo otimização por enxame de partículas como o de melhor performance em termos da qualidade das soluções finais.

Uma problemática no emprego dos algoritmos *soft-computing* em [27] e [28] consiste na escolha de parâmetros internos desses algoritmos. Dessa forma, em [29], uma análise exploratória estatística dos algoritmos *soft-computing* foi realizada, a qual permitiu a obtenção de parâmetros ótimos para os algoritmos de resolução do problema multi-objetivo em (3.33). Ainda nesse trabalho, empregou-se ferramentas estatísticas de projeto de experimentos para realização de comparações de performance entre os algoritmos *soft-computing*. Novamente, confirmando os resultados obtidos em [28], o algoritmo otimização por enxame de partículas obteve melhor performance e dessa forma, fortes evidências estatísticas confirmam que esse algoritmo é a melhor escolha na resolução do problema multi-objetivo de projeto de filtros digitais FIR de fase não-linear [29].

O presente trabalho emprega a metodologia de projeto de filtros digitais desenvolvida em [29]. Particularmente, serão utilizados algoritmos *soft-computing* para resolução do problema multi-objetivo em (3.33), conforme apresentados no Capítulo 5. Adicionalmente, também serão utilizados algoritmos baseados em informação do gradiente, mais especificamente o algoritmo Quasi-Newton, conforme discussão detalhada no Capítulo 4. No Capítulo 6 será desenvolvida

a metodologia exploratória estatística para obtenção dos parâmetros ótimos dos algoritmos *soft-computing* e de ferramentas estatísticas que possibilitem comparações de performance entre os vários algoritmos de otimização empregados na literatura. No Capítulo 7, onde for possível, comparações à nível da performance entre as várias abordagens de resolução do problema multi-objetivo na literatura serão feitas. Convém, por fim notar que, nenhuma consideração sobre a globalidade do conjunto Pareto-ótimo será feita no presente trabalho.

# 4 ALGORITMOS DE OTIMIZAÇÃO NÃO-LINEAR BASEADOS NO GRADIENTE

## 4.1 INTRODUÇÃO

A família de algoritmos apresentados nesse capítulo tem como elemento fundamental na sua composição, informação do gradiente  $\nabla f$  da função objetivo  $f$ . Mais especificamente, os algoritmos de resolução do problema de otimização em (3.5) apresentados nesse capítulo fazem uso de condições de primeira e segunda ordem para determinar o mínimo local da função objetivo  $f$ . Tais condições são expressas pelo teorema de Taylor, uma ferramenta matemática usada no estudo de minimizadores de funções suaves [14].

Antes de prosseguir com uma discussão detalhada dos métodos de otimização, se faz necessário definir alguns termos de uso comum na literatura. Primeiramente, uma função é dita suave se a mesma é contínua e diferenciável [10]. Todos os algoritmos da família de métodos de otimização baseados no gradiente partem do princípio que a função objetivo é suave. Na derivação dos métodos, assume-se que a função  $f$  é duas vezes continuamente diferenciável, embora de fato, uma suposição mínima é que  $f$  tenha apenas o gradiente contínuo [17].

O gradiente de uma função suave  $f(\mathbf{x})$  é definido como o vetor de derivadas parciais de primeira ordem em um ponto  $\mathbf{x} \in \mathbb{R}^n$  [10]:

$$\nabla f(\mathbf{x}) = \begin{pmatrix} \partial f / \partial x_1 \\ \partial f / \partial x_2 \\ \vdots \\ \partial f / \partial x_n \end{pmatrix} \quad (4.1)$$

Se  $f(\mathbf{x})$  é duas vezes continuamente diferenciável, então existe uma matriz de derivadas parciais de segunda ordem chamada de matriz Hessiana [10]:

$$\nabla^2 f(\mathbf{x}) = \begin{vmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{vmatrix} \quad (4.2)$$

Outro termo importante é o conceito de linha  $\ell$  em um espaço  $n$ -dimensional, ou seja, o



conjunto de todos os pontos que obedecem a seguinte relação no espaço  $\mathbb{R}^n$  [10]:

$$\ell(\alpha) = \mathbf{x}' + \alpha \mathbf{p}, \quad (4.3)$$

para todo  $\alpha$  no qual  $\mathbf{x}'$  é um ponto fixo, correspondendo a  $\alpha = 0$ , ao longo da linha  $\ell$  e  $\mathbf{p}$  é a direção da linha. Esse conceito será de suma importância na derivação dos métodos de busca em linha, considerados como a espinha dorsal dos algoritmos de otimização baseados no gradiente. Nesses métodos, resolve-se o problema de minimização uni-dimensional através da linha  $\ell$ , obtendo-se o  $\alpha$  com o menor valor funcional em  $f(\mathbf{x} + \alpha \mathbf{p})$ . A ilustração de uma linha  $\ell$  no espaço  $\mathbb{R}^2$  pode ser vista na Figura 4.1.

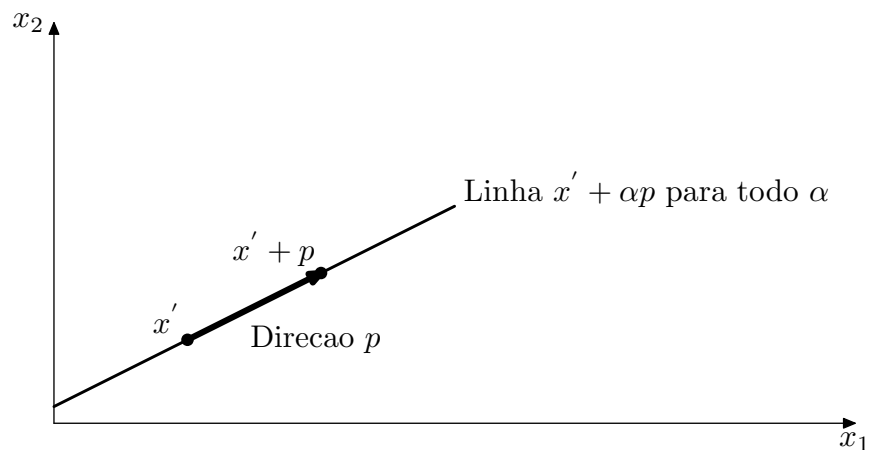


Figura 4.1: Linha  $\ell$  com direção de busca  $\mathbf{p}$  em duas dimensões

De forma geral, a família de algoritmos de otimização não-linear baseados no gradiente possuem a seguinte estrutura [10]:

---

**Algoritmo 4.1.1:** Estrutura geral de algoritmos de otimização não-linear baseados no gradiente

---

- 4.1.1.1 Determine uma direção de busca  $\mathbf{p}_k$ ;
  - 4.1.1.2 Ache  $\alpha_k$  que minimize  $f(\mathbf{x}_k + \alpha \mathbf{p}_k)$  com respeito a  $\alpha$ ;
  - 4.1.1.3  $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k \mathbf{p}_k$ ;
- 

Todos os algoritmos da família de métodos baseados no gradiente necessitam que o usuário com o conhecimento da área de aplicação do algoritmo, ou seja, da função objetivo  $f$ , forneça uma posição inicial denotada por  $\mathbf{x}_0$ . Caso não se possua tal conhecimento *a priori* de  $f$ , uma estimativa inicial  $\mathbf{x}_0$  pode ser escolhida pelo algoritmo, através de uma maneira sistemática ou de alguma forma arbitrária [14].

Convém notar ainda com relação ao algoritmo 4.1.1 que diferentes métodos de otimização da família de algoritmos baseados no gradiente correspondem à diferentes maneiras de geração da direção de busca  $\mathbf{p}_k$ .

Uma possível alternativa de geração da direção de busca no algoritmo 4.1.1 consiste na direção de maior descendência, ou seja, através da direção  $\mathbf{p}_k = -\nabla f_k$  em cada passo do algoritmo. Essa metodologia de geração da direção de busca  $\mathbf{p}$  deu surgimento ao algoritmo do gradiente máximo. Uma vantagem desse método se deve ao fato que apenas é necessário o cálculo do gradiente, dispensando assim o uso de derivadas de segunda ordem [14].

Outro importante método de geração da direção de busca consiste no método de Newton. A direção de busca  $\mathbf{p}_k$  desse método é derivada da aproximação da série de Taylor de segunda ordem para  $f(\mathbf{x}_k + \mathbf{p})$ , gerando o seguinte modelo quadrático  $m_k(\mathbf{p})$  [14]:

$$m_k(\mathbf{p}) \triangleq f(\mathbf{x}_k + \mathbf{p}) \approx f_k + \mathbf{p}^T \nabla f_k + \frac{1}{2} \mathbf{p}^T \nabla^2 f_k \mathbf{p} \quad (4.4)$$

A partir do modelo quadrático  $m_k(\mathbf{p})$  obtém-se a direção de Newton achando o vetor  $\mathbf{p}$  que minimiza  $m_k(\mathbf{p})$ . Para isso, iguala-se a derivada de  $m_k(\mathbf{p})$  a zero, obtendo a seguinte fórmula explícita [14]:

$$\mathbf{p}_k = -(\nabla^2 f_k)^{-1} \nabla f_k \quad (4.5)$$

Geralmente, o método de Newton possui uma rápida convergência local. Sua principal desvantagem consiste na necessidade de obtenção da matriz Hessiana  $\nabla^2 f(\mathbf{x})$ . A codificação explícita dessa matriz em programas computacionais pode ser custosa e propensa a erros [14].

Uma alternativa interessante ao método de Newton consiste nos métodos Quasi-Newton. Esses métodos não necessitam utilizar a matriz Hessiana  $\nabla^2 f(\mathbf{x})$  e ainda sim possuem rápida convergência local. Ao invés da matriz Hessiana, usa-se uma aproximação  $\mathbf{B}_k$ , a qual é atualizada após cada passo do algoritmo de otimização e leva em conta conhecimento adicional ganho durante cada passo do algoritmo [14]. Sendo assim, a fórmula de geração da nova direção de busca no método Quasi-Newton é definida por:

$$\mathbf{p}_k = -\mathbf{B}_k^{-1} \nabla f_k \quad (4.6)$$

Entre os diferentes métodos de atualização da matriz de aproximação  $\mathbf{B}_k$  pode-se citar o método BFGS (Broyden-Fletcher-Goldfarb-Shanno) e o método DFP (Davidon-Fletcher-Powell).

Convém, por fim, discutir uma última maneira de geração de  $\mathbf{p}_k$ , através de métodos de gradiente conjugado não-lineares. Esses métodos possuem a seguinte fórmula de geração de  $\mathbf{p}_k$  [14]:

$$\mathbf{p}_k = -\nabla f(\mathbf{x}_k) + \beta_k \mathbf{p}_{k-1}, \quad (4.7)$$

onde  $\beta_k$  é um escalar que garante que  $\mathbf{p}_k$  e  $\mathbf{p}_{k-1}$  sejam conjugados, um conceito importante na minimização de funções quadráticas [14].

Em geral, os métodos de gradiente conjugado são melhores que o método do gradiente máximo e são quase tão simples de se implementar computacionalmente, além de não necessitarem do armazenamento de matrizes. Entretanto, esses métodos não possuem a mesma taxa de convergência que os métodos de Newton e Quasi-Newton [14].

Em resumo, como regra geral, para ser eficiente, um método de otimização baseado no gradiente deve fazer uso de condições de segunda ordem, tais como os métodos Newton e Quasi-Newton. Com relação aos métodos do gradiente conjugado, atualmente o mesmo se encontra em desuso na comunidade de otimização, além de possuir aplicação restrita apenas a problemas especializados, sendo hoje substituído quase que exclusivamente pelos métodos Quasi-Newton em problemas de otimização não-lineares e de difícil resolução [17].

Com relação à aplicação dos métodos de otimização baseados no gradiente apresentados nesse capítulo ao problema de aproximação arbitrária de módulo e fase no projeto de filtros digitais de fase não-linear, apresentado na seção 2.3 do Capítulo 2, convém notar que nesse trabalho será aplicado apenas o método Quasi-Newton na minimização da função objetivo em (2.47). Isso se deve principalmente à performance e robustez do método em comparação com os demais métodos de otimização baseados no gradiente. Ainda assim, os demais métodos de otimização serão discutidos com relativa profundidade, mas apenas com fim didático. Particularmente, com relação ao método do gradiente conjugado, se faz oportuno apresentá-lo de forma mais extensa devido à sua possível aplicação ao projeto de filtros FIR de fase não-linear de alta ordem, mesmo que essa aplicação não seja analisada no Capítulo 7.

O restante desse capítulo é organizado da seguinte forma. A seção 4.2 apresenta as condições de primeira e segunda ordem derivadas a partir do teorema de Taylor assim como a ligação dessas condições com a obtenção de minimizadores locais da função objetivo  $f$ . Já a seção 4.3 apresenta discussão detalhada dos algoritmos de busca em linha para minimização unidimensional em linha. As seções 4.4, 4.5, 4.6 e 4.7 apresentam respectivamente, discussão detalhada dos métodos do gradiente máximo, gradiente conjugado, método de Newton e método Quasi-Newton. Por fim, como referência, a definição formal dos algoritmos derivados nesse capítulo encontra-se no apêndice I.

## **4.2 SOLUÇÕES LOCAIS E CONDIÇÕES DE PRIMEIRA E SEGUNDA ORDEM**

Conforme mencionado no início do presente capítulo, a ferramenta matemática utilizada para estudo de minimizadores de funções suaves é representada pelo teorema de Taylor. A partir desse importante teorema, pode-se estabelecer as condições necessárias e suficientes de um minimizador local  $\mathbf{x}^*$ . Além disso, a derivação teórica das diferentes técnicas de geração da nova direção de busca  $\mathbf{p}$  para a família de algoritmos baseados no gradiente toma como base as implicações desse teorema.

Sendo assim, para apresentar o teorema, primeiramente apresenta-se a função  $f(\mathbf{x} + \alpha \mathbf{p})$  por meio de sua representação pela série infinita de Taylor[10]:

$$f(\mathbf{x} + \alpha \mathbf{p}) = f(\mathbf{x}) + \alpha \mathbf{p}^T \nabla f(\mathbf{x}) + \frac{1}{2} \alpha^2 \mathbf{p}^T \nabla^2 f(\mathbf{x}) \mathbf{p} + \dots \quad (4.8)$$

O teorema de Taylor contempla então a aproximação de primeira e segunda ordem da série em (4.8):

**Teorema de Taylor 4.2.1** *Suponha que  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  seja continuamente diferenciável e que  $\mathbf{p} \in \mathbb{R}^n$ . Então tem-se a seguinte aproximação de primeira ordem:*

$$f(\mathbf{x} + \mathbf{p}) = f(\mathbf{x}) + \nabla f(\mathbf{x} + t\mathbf{p})^T \mathbf{p}, \quad (4.9)$$

*para algum  $t \in (0, 1)$ . Ainda, se  $f$  é duas vezes continuamente diferenciável tem-se a seguinte aproximação de segunda ordem:*

$$f(\mathbf{x} + \mathbf{p}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T \nabla^2 f(\mathbf{x} + t\mathbf{p}) \mathbf{p}, \quad (4.10)$$

*para algum  $t \in (0, 1)$ .*

A partir da definição do teorema de Taylor, pode se estabelecer as condições necessárias e suficientes de um minimizador local  $\mathbf{x}^*$ . Primeiramente, a condição necessária de primeira ordem é postulada por [14]:

**Condição necessária de primeira ordem 4.2.1** *Se  $\mathbf{x}^*$  é um minimizador local e  $f$  é continuamente diferenciável em um intervalo aberto de  $\mathbf{x}^*$ , então  $\nabla f(\mathbf{x}^*) = 0$ .*

Em [14] prova-se a condição necessária de primeira ordem através do teorema de Taylor em (4.9). Convém ainda notar que o minimizador local  $\mathbf{x}^*$  é um ponto estacionário já que  $\nabla f(\mathbf{x}^*) = 0$ .

A condição necessária de segunda ordem é postulada por [14]:

**Condição necessária de segunda ordem 4.2.1** *Se  $\mathbf{x}^*$  é um minimizador local de  $f$  e  $\nabla^2 f$  existe e é contínua em um intervalo aberto de  $\mathbf{x}^*$ , então  $\nabla f(\mathbf{x}^*) = 0$  e  $\nabla^2 f(\mathbf{x}^*)$  é positiva semi-definida.*

Na condição necessária de segunda ordem acima, a matriz Hessiana  $\mathbf{H} = \nabla^2 f$  é positiva semi-definida se  $\mathbf{p}^T \mathbf{H} \mathbf{p} \geq 0$ . Novamente, em [14] prova-se a condição necessária de segunda ordem através do teorema de Taylor em (4.10).

Por fim, é conveniente definir a condição suficiente de segunda ordem que garante  $\mathbf{x}^*$  como um minimizador local estrito [14]:

**Condição suficiente de segunda ordem 4.2.1** *Suponha que  $\nabla^2 f$  é contínua em um intervalo aberto de  $\mathbf{x}^*$  e que  $\nabla f(\mathbf{x}^*) = 0$  e  $\nabla^2 f(\mathbf{x}^*)$  é positiva definida. Então  $\mathbf{x}^*$  é um minimizador local estrito de  $f$ .*

, onde na condição suficiente de segunda ordem acima, a matriz Hessiana  $\mathbf{H} = \nabla^2 f$  é positiva definida se  $\mathbf{p}^T \mathbf{H} \mathbf{p} > 0$ .

As condições apresentadas nessa seção oferecem a fundação teórica para os algoritmos de otimização baseados no gradiente. De uma maneira ou de outra, todos os algoritmos procuram um ponto  $\mathbf{x}$  onde seu gradiente  $\nabla f(\mathbf{x})$  diminua [14].

### 4.3 ALGORITMOS DE BUSCA EM LINHA

Em um método de busca em linha, é fornecida uma função contínua e diferenciável  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  e uma direção descendente  $\mathbf{p}$  para  $f$  num dado ponto  $\mathbf{x} \in \mathbb{R}^n$ . Considere a seguinte função auxiliar:

$$\phi(\alpha) \equiv f(\mathbf{x} + \alpha \mathbf{p}), \quad \alpha \geq 0 \quad (4.11)$$

Algoritmos de busca em linha procuram então por tamanhos de passo  $\alpha$  em (4.11) que obedecem certos critérios.

O algoritmo 4.1.1 apresentado anteriormente é idealizado, já que a minimização de  $\phi(\alpha)$ , em geral, não pode ser implementada em um número finito de operações [10]. Na prática, é necessário terminar o algoritmo quando certas condições são satisfeitas para um mínimo aproximado na direção de busca  $\mathbf{p}_k$ . Adicionalmente, como pode não ser eficiente minimizar, com alta precisão  $\phi(\alpha)$ , para situações onde  $\mathbf{x}$  esteja remotamente distante da posição mínima global  $\mathbf{x}^*$ , tais condições são escolhidas de forma à serem prontamente satisfeitas. Nessas circunstâncias, o algoritmo emprega uma busca parcial ou inexata.

Uma condição bastante utilizada para uma busca inexata no algoritmo de busca em linha são as condições de Wolfe, as quais obedecem os seguintes critérios:

$$\phi(\alpha) \leq \phi(0) + \mu \phi'(0) \alpha \quad (4.12)$$

$$\left| \phi'(\alpha) \right| \leq \eta \left| \phi'(0) \right|, \quad (4.13)$$

com  $\phi'(\alpha)$  definido por:

$$\phi'(\alpha) = \nabla f(\mathbf{x} + \alpha \mathbf{p})^T \mathbf{p} \quad (4.14)$$

A condição (4.12) é conhecida como condição de decrescimento suficiente, já que a mesma força o decrescimento do valor funcional de (4.11). Entretanto, somente essa condição não é suficiente para convergência garantida do método, já que a mesma permite escolhas arbitrariamente pequenas de  $\alpha > 0$  [30]. A condição (4.13) evita escolhas suficientemente pequenas de  $\alpha$  e é conhecida como condição de curvatura. Em particular, essa condição deve ser atendida para garantir uma atualização positiva definida da matriz Hessiana no método Quasi-Newton. Adicionalmente, para métodos de otimização por gradiente conjugado, atendendo as condições de Wolfe, consegue-se gerar sempre uma direção descendente [14].

Na prática, o valor de  $\mu$  na condição de decrescimento suficiente (4.12) possui valor bastante reduzido, de tal forma que dificilmente mais de um decrescimento funcional seja necessário. Em [31], sugere-se então o valor de  $\mu = 1 \times 10^{-4}$  para atender esse requisito. Essa sugestão do valor de  $\mu$  também é seguida na implementação do algoritmo de busca em linha desse trabalho. Com relação ao valor  $\eta$  na condição de curvatura, geralmente seu valor é escolhido de acordo com o método de geração da nova direção de busca  $\mathbf{p}$ . Valores típicos são de  $\eta = 0.9$  quando a direção de busca é gerada pelo método de Newton ou pelo método Quasi-Newton e  $\eta = 0.1$  quando a direção é gerada por um método do gradiente conjugado [14].

Um procedimento comum nos algoritmos de busca em linha é o de primeiramente verificar se o tamanho de passo inicial  $\alpha_k = 1$  atende à condição de decrescimento suficiente (4.12), e se não atende-la, reduzir sistematicamente o intervalo de busca na direção  $\mathbf{p}_k$ . A redução do intervalo no método de busca em linha que será apresentado abaixo segue a metodologia em [31], onde a condição de curvatura (4.13) é satisfeita pela rejeição de tamanhos de passo excessivamente pequenos na estratégia de redução do intervalo de busca [31].

A estrutura geral de um algoritmo de busca em linha com redução do sistemática do intervalo, que atende as condições de Wolfe (4.12) e (4.13), pode ser vista pelo conjunto sistemático de passos do algoritmo 4.3.1 [31].

---

**Algoritmo 4.3.1:** Estrutura geral de um algoritmo de busca em linha

---

- 4.3.1.1 Dado  $\mu \in (0, \frac{1}{2})$ ,  $0 < l < u < 1$ ;  
 4.3.1.2  $\alpha_k = 1$ ;  
 4.3.1.3 **enquanto**  $\phi(\alpha_k) > \phi(0) + \mu\phi'(0)\alpha_k$  **faça**  
 4.3.1.4      $\alpha_k \leftarrow \rho\alpha_k$  para algum  $\rho \in [l, u]$ ;  
 4.3.1.5      $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k\mathbf{p}_k$
- 

Para a estratégia de redução de  $\alpha_k$  utiliza-se a abordagem de usar a informação corrente sobre  $\phi(\alpha_k)$ , criar um modelo funcional, e achar o próximo valor de  $\alpha$  que minimize esse modelo. Inicialmente, possuem-se duas informações sobre  $\phi(\alpha)$ ,  $\phi(0) = f(\mathbf{x}_k)$  e  $\phi'(0) = \nabla f(\mathbf{x}_k)^T \mathbf{p}_k$ . Após calcular  $f(\mathbf{x}_k + \mathbf{p}_k)$ , possui-se também,  $\phi(1) = f(\mathbf{x}_k + \mathbf{p}_k)$ , de tal forma que, se  $f(\mathbf{x}_k + \mathbf{p}_k)$  não satisfaz a condição de decrescimento suficiente (4.12), constrói-se um modelo de  $\phi(\alpha)$  pelo quadrático unidimensional [31]:

$$m_q(\alpha) = \left[ \phi(1) - \phi(0) - \phi'(1) \right] \alpha^2 + \phi'(0) \alpha + \phi(0) \quad (4.15)$$

Minimizando esse modelo quadrático,  $m_q(\alpha^*) = 0$ , obtém-se:

$$\alpha^* = \frac{-\phi'(0)}{2[\phi(1) - \phi(0) - \phi'(0)]} \quad (4.16)$$

Adicionalmente, se o valor de  $\phi(1)$  é muito maior que  $\phi(0)$ ,  $\alpha$  pode ser bastante pequeno. Nessa situação, não é desejável diminuir  $\alpha$  demasiadamente, já que provavelmente, nesse caso  $\phi(\alpha)$  é pobremente modelado por essa região quadrática, e impõe-se um intervalo inferior de  $l = \frac{1}{10}$  no algoritmo 4.3.1 [31].

Após a primeira iteração do laço do algoritmo 4.3.1, suponha que  $\phi(\alpha_k)$  ainda não satisfaça a condição de decrescimento suficiente (4.12). Nesse caso, é necessário um nova redução do intervalo de busca. Embora seja possível o uso do modelo quadrático (4.15), no laço corrente do algoritmo, possuem-se quatro informações sobre  $\phi(\alpha)$ . Dessa forma, nesse laço, e nos laços seguintes, utiliza-se um modelo cúbico de  $\phi$ ,  $m_{cub}(\alpha)$  ajustando esse modelo a  $\phi(0)$ ,  $\phi'(0)$  e os dois valores anteriores de  $\phi(\alpha)$ . A razão para o uso do modelo cúbico se deve ao fato de que tal modelo pode modelar mais precisamente situações onde  $f$  tenha curvatura negativa. Além do mais, tal modelo cúbico possui um minimizador local único [31].

O cálculo do modelo cúbico  $m_{cub}(\alpha)$  é feito da seguinte forma. Definindo  $\alpha_{ant}$  e  $\alpha_{2ant}$  como os dois últimos valores de  $\alpha_k$ , o modelo cúbico que ajusta  $\phi(0)$ ,  $\phi'(0)$ ,  $\phi(\alpha_{ant})$  e  $\phi(\alpha_{2ant})$  é definido por:

$$m_{cub}(\alpha) = a\alpha^3 + b\alpha^2 + \phi'(0)\alpha + \phi(0), \quad (4.17)$$

onde:

$$\begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{\alpha_{ant} - \alpha_{2ant}} \begin{bmatrix} \frac{1}{\alpha_{ant}^2} & \frac{-1}{\alpha_{2ant}^2} \\ -\frac{\alpha_{2ant}}{\alpha_{ant}^2} & \frac{\alpha_{ant}}{\alpha_{2ant}^2} \end{bmatrix} \begin{bmatrix} \phi(\alpha_{ant}) - \phi(0) - \phi'(0)\alpha_{ant} \\ \phi(\alpha_{2ant}) - \phi(0) - \phi'(0)\alpha_{2ant} \end{bmatrix} \quad (4.18)$$

Felizmente, o minimizador local  $\alpha^*$  desse modelo cúbico possui fórmula fechada, a qual é expressa pela seguinte relação [31]:

$$\alpha^* = \frac{-b + \sqrt{b^2 - 3a\phi'(0)}}{3a} \quad (4.19)$$

A definição formal detalhada do algoritmo de busca em linha apresentado nessa seção encontra-se no apêndice I. O algoritmo I.1.6 contém o corpo principal da rotina de minimização. Por sua vez, essa rotina faz a chamada das sub-rotinas auxiliares I.1.7 e I.1.8. Rapidamente, o algoritmo

I.1.6 contém o laço principal do método de busca em linha. Já o algoritmo I.1.7 constrói o modelo quadrático (4.15) ou cúbico (4.17) para encontrar um  $\alpha$  que minimize esse modelo e atenda a condição de decrescimento suficiente (4.12). Por fim, o algoritmo I.1.8 gera um valor de  $\alpha$  que atenda a condição de curvatura (4.13).

## 4.4 MÉTODO DO GRADIENTE MÁXIMO

A geração da direção de busca com o método do gradiente máximo pode ser derivada a partir do teorema de Taylor apresentado na seção 4.2. Para isso, considere a aproximação de segunda ordem da série de Taylor em (4.10) em função do valor de  $\alpha$ , a qual diz que para qualquer direção de busca  $\mathbf{p}$  e valor de  $\alpha$ , tem-se a seguinte relação [14]:

$$f(\mathbf{x}_k + \alpha \mathbf{p}) = f(\mathbf{x}_k) + \alpha \mathbf{p}^T \nabla f_k + \frac{1}{2} \alpha^2 \mathbf{p}^T \nabla^2 f(\mathbf{x}_k + t\mathbf{p}) \mathbf{p} \quad \text{para algum } t \in (0, \alpha) \quad (4.20)$$

Com base na aproximação de segunda ordem em (4.20), a taxa de mudança em  $f$  ao longo da direção  $\mathbf{p}$  em  $\mathbf{x}_k$  é simplesmente o coeficiente de  $\alpha$ , ou seja,  $\mathbf{p}^T \nabla f_k$  [14]. Assim, a direção unitária  $\mathbf{p}$  de decrescimento mais rápido é obtida a partir da resolução do seguinte problema de otimização com restrição [14]:

$$\min_{\mathbf{p}} \mathbf{p}^T \nabla f_k, \quad \text{sujeito a} \quad \|\mathbf{p}\| = 1, \quad (4.21)$$

onde o operador  $\|\cdot\|$  denota a norma  $\ell_1$ :

$$\|\mathbf{x}\|_1 \triangleq \sum_{i=1}^n \|\mathbf{x}_i\| \quad (4.22)$$

A expressão para minimização em (4.21),  $\mathbf{p}^T \nabla f_k$ , pode ser expressa pela seguinte relação [14]:

$$\mathbf{p}^T \nabla f_k = \|\mathbf{p}\| \|\nabla f_k\| \cos \theta = \|\nabla f_k\| \cos \theta, \quad (4.23)$$

onde  $\theta$  é o ângulo entre  $\mathbf{p}$  e  $\nabla f_k$ . Claramente, o minimizador da expressão em (4.23) é obtido quando  $\cos \theta = -1$  e dessa forma:

$$\mathbf{p} = -\frac{1}{\|\nabla f_k\|} \nabla f_k \quad (4.24)$$

Como observação final, deve-se dizer que o método do gradiente máximo não deve ser usado como minimizador do problema de otimização não-linear sem restrição na maioria das circuns-



tâncias, já que outros algoritmos como o método Quasi-Newton, por exemplo, são bem mais eficientes [31].

## 4.5 MÉTODOS COM GRADIENTE CONJUGADO

Diferentemente dos métodos que utilizam informação de derivadas de primeira e segunda ordem (método de Newton, seção 4.6) ou informação da derivada de primeira ordem e uma estimativa de segunda ordem (método Quasi-Newton, seção 4.7), uma metodologia alternativa baseada no modelo de segunda ordem faz uso da propriedade de conjugação de um conjunto de vetores com elementos diferentes de zero,  $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n$ , para uma dada matriz positiva definida  $\mathbf{H}$ . Essa é a propriedade na qual a seguinte relação é válida [10]:

$$\mathbf{p}_i^T \mathbf{H} \mathbf{p}_j = 0 \quad \forall i \neq j \quad (4.25)$$

De particular importância, os métodos com gradiente conjugado usam a idéia de conjugação (4.25) aliada ao método do gradiente máximo. Tais métodos serão o objeto de estudo dessa seção.

A apresentação desses métodos segue a abordagem de [10], onde primeiramente deriva-se a técnica do gradiente conjugado com aplicação prática somente em funções quadráticas. Posteriormente, o método é modificado para sua utilização em outras funções não-lineares. Convém notar ainda que, várias variantes do método gradiente conjugado existem na literatura de otimização. A primeira forma que será apresentada consiste no método de Fletcher-Reeves, historicamente o primeiro algoritmo com gradiente conjugado não-linear a aparecer na literatura. Outra forma que será apresentada nessa seção consiste no método Polak-Ribiere.

Os métodos com gradiente conjugado tentam associar as propriedades de conjugação (4.25) com o método do gradiente máximo em uma tentativa de se alcançar eficiência e confiabilidade [10]. Com fim didático, apresenta-se primeiramente o método de maneira idealizada, já que o método nessa forma possui aplicação prática somente em funções quadráticas. Para isso, faça a direção de busca na primeira iteração do algoritmo ser expressa como a direção gerada pelo método do gradiente máximo,  $\mathbf{p}_1 = -\nabla f_1$  e nas iterações seguintes  $k \geq 1$ , pela seguinte relação [10]:

$$\mathbf{p}_{k+1} = \text{componente de } -\nabla f_{k+1} \text{ conjugado a } \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k \quad (4.26)$$

Pode-se utilizar do processo Gram-Schmidt <sup>1</sup> para expressar a próxima  $k$ -ésima direção de busca  $\mathbf{p}_{k+1}$  por [10]:

---

<sup>1</sup>Cada espaço linear de dimensão finita tem uma base finita. Se esse espaço é Euclidiano, pode-se sempre construir uma base ortogonal. Esse resultado é deduzido como consequência de um teorema geral no qual a prova mostra como se construir conjuntos ortogonais em qualquer espaço Euclidiano, de dimensão finita ou infinita. Essa construção é chamada de processo de ortogonalização de Gram-Schmidt. [13]

$$\mathbf{p}_{k+1} = -\nabla f_{k+1} + \sum_j \beta_j \mathbf{p}_j \quad (4.27)$$

Para funções quadráticas,  $\beta_j = 0$  para todo  $j < k$  e a expressão simplifica-se a [10]:

$$\mathbf{p}_{k+1} = -\nabla f_{k+1} + \beta_k \mathbf{p}_k, \quad (4.28)$$

com  $\beta_0 = 0$ , onde:

$$\beta_k = \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k} \quad (4.29)$$

A fórmula de atualização de  $\beta_k$  em (4.29) é conhecida como o método de Fletcher-Reeves, desenvolvida diretamente do método com gradiente conjugado de Hestenes e Stiefel para resolução de sistemas lineares [10].

Uma possível desvantagem do método Fletcher-Reeves em (4.29) é que a importante propriedade de descendência  $\mathbf{p}_k^T \nabla f_k \leq 0$  pode nem sempre ser mantida na geração da nova direção de busca  $\mathbf{p}_k$  [10]. Entretanto, é possível preservar a propriedade de descendência (para funções quadráticas) e ainda sim usar uma simples fórmula de atualização do parâmetro  $\beta_k$  conforme a seguinte relação [10]:

$$\beta_k = \frac{(\nabla f_{k+1} - \nabla f_k)^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k} \quad (4.30)$$

A atualização do parâmetro  $\beta_k$  em (4.30) é também conhecida como método de Polak-Ribiere.

Tendo apresentado as duas fórmulas de atualização do parâmetro  $\beta_k$  em (4.29) e (4.30), se faz necessário considerar como o método do gradiente conjugado em (4.28) pode ser aplicado a minimização de funções não-quadráticas. A primeira observação com relação à essa aplicação refere-se a qual tipo de algoritmo de busca em linha pode ser utilizado e quão preciso esse algoritmo deve ser [10]. Como se espera que o gradiente conjugado seja um método que gere novas direções descendentes de busca  $\mathbf{p}$  e como buscas em linha precisas funcionam bem nesse método, o algoritmo de busca em linha apresentado na seção 4.3, o qual obedece as condições de Wolfe (4.12) e (4.13), conjugado à uma busca precisa, para isso, atribuindo  $\eta = 0.1$ , é altamente recomendado [10].

A segunda observação com relação a aplicação do método em funções não-quadráticas surge devido a iteração do método não mais terminar em um número finito de passos para funções não-quadráticas [10]. Para combater isso, existe então a possibilidade de reiniciar  $\mathbf{p}_k$  periodicamente com a direção do gradiente máximo.

A estratégia mais popular para reiniciar a direção de busca  $\mathbf{p}$  faz uso da observação de que os gradientes são mutuamente ortogonais quando a função objetivo  $f$  é quadrática. Dessa forma,

reinicia-se  $\mathbf{p}$  com a direção do gradiente máximo, ou seja, fazendo  $\beta_k = 0$  em (4.28), sempre quando dois gradientes consecutivos são distantes ortogonalmente, ou seja, através do seguinte teste [14]:

$$\frac{|\nabla f_k^T \nabla f_{k-1}|}{\|\nabla f_k\|^2} \geq \nu, \quad (4.31)$$

onde um valor típico para o parâmetro  $\nu$  em (4.31) é de  $\nu = 0.1$ .

Uma aparente vantagem da estratégia acima é que se a iteração progredir de uma região não-quadrática em uma vizinhança da solução na qual a função objetivo  $f$  é bem aproximada por um modelo quadrático, então espera-se que a estratégia de reiniciar a direção de busca  $\mathbf{p}$  com a direção do gradiente máximo possua rápida convergência [10].

Por fim, convém notar que quando comparado, à nível da qualidade da solução final  $\mathbf{x}^*$ , com o método Quasi-Newton, fica claro que o método do gradiente conjugado é ao mesmo tempo menos eficiente e robusto e não deve ser utilizado no problema de otimização não-linear sem restrição em circunstâncias normais [10]. Entretanto, existe uma vantagem particular nesses métodos, a forma particularmente simples de geração da nova direção de busca  $\mathbf{p}_{k+1}$  em (4.28), a qual não necessita de operações matriciais. De fato, o método de Fletcher-Reeves necessita apenas do armazenamento de três vetores de dimensão  $n$  para sua implementação, enquanto que o método de Polak-Ribiere necessita de quatro. Dessa forma, os métodos com gradiente conjugado podem ser a única classe de algoritmos baseados no gradiente aplicáveis a problemas de larga-escala [10].

## 4.6 MÉTODO DE NEWTON

Conforme brevemente discutido na seção 4.1, a geração da nova direção de busca  $\mathbf{p}_k$  no método de Newton, equação (4.5), é obtida a partir da minimização do modelo quadrático  $m_k(\mathbf{p})$  em (4.4). Tal modelo quadrático, por sua vez, foi obtido através do truncamento da expansão da série de Taylor de  $f(\mathbf{x})$  sobre  $\mathbf{x}_k$ .

Convém notar que, diferentemente do método do gradiente máximo, usa-se por padrão o valor de  $\alpha = 1$  associado à direção de Newton no método de busca em linha. A maioria das implementações do método de Newton usam o tamanho de passo unitário  $\alpha = 1$  onde possível e ajustam  $\alpha$  somente quando o mesmo não produz uma redução satisfatória no valor de  $f$  [14].

Embora o método de Newton possua rápida convergência local, o mesmo apresenta dificuldade de convergência quando a matriz Hessiana  $\mathbf{H}$  não é positiva-definida. Na maioria dos casos ainda é possível gerar a nova direção de busca  $\mathbf{p}_k$  a partir de (4.5) e fazer uma busca em linha ao longo de  $\pm \mathbf{p}_k$ , sendo que o sinal é escolhido para garantir uma direção descendente [10]. Entretanto, o ponto estacionário da função quadrática de aproximação na série de Taylor de segunda

ordem,  $\nabla f(\mathbf{x}^*) = 0$ , não é um ponto de minimização, e assim, a relevância de pesquisar nessa direção torna-se questionável [10].

Claramente, alguma modificação adicional é necessária em (4.5) para se determinar um método de aplicação generalizada. Uma primeira possibilidade consiste em reverter a direção do gradiente máximo  $\mathbf{p}_k = -\nabla f_k$  sempre que a matriz Hessiana  $\mathbf{H}$  não for positiva-definida. Entretanto, se essa modificação opera sucessivamente em um certo número de iterações do algoritmo de otimização, então é provável que o comportamento oscilatório lento do método do gradiente máximo seja observado. Esse fenômeno ocorre devido a informação no modelo quadrático  $m_k(\mathbf{p})$  em (4.4) ser ignorada [10].

Brevemente, uma segunda possibilidade, e talvez a mais utilizada para se atacar o problema de convergência quando a matriz Hessiana  $\mathbf{H}$  não é positiva definida, consiste em modificar a nova direção de busca do método de Newton em (4.5) através de um novo modelo quando assim for necessário, de tal forma que esse modelo tenha um minimizador único. Pode-se utilizar então desse modelo, para definir a nova direção de busca [31]. Isso é mais convenientemente alcançado adicionando um múltiplo da matriz identidade à matriz Hessiana  $\mathbf{H}_k$  e resolvendo o sistema linear [10]:

$$(\mathbf{H}_k + \mu\mathbf{I}) \mathbf{p}_k = -\nabla f_k \quad (4.32)$$

Resultando na nova direção de busca modificada de Newton [31]:

$$\mathbf{p}_k = -(\mathbf{H}_k + \mu\mathbf{I})^{-1} \nabla f_k, \quad (4.33)$$

onde  $\mu \geq 0$  e  $(\mathbf{H} + \mu\mathbf{I})$  é positiva-definida. Dessa forma, essa modificação na direção de busca usa em grande parte informação quadrática mesmo quando a matriz Hessiana  $\mathbf{H}_k$  não é positiva-definida, e como consequência, torna-se mais efetiva do que a primeira possibilidade mencionada acima [10].

Com relação a aplicação prática do método de Newton ao problema de otimização no projeto de filtros digitais FIR de fase não-linear, faz-se lembrar que o método necessita da codificação em programas computacionais da matriz Hessiana  $\mathbf{H}_k$ , a qual impõe obter expressões fechadas das derivadas parciais de segunda ordem de (2.47). Conforme visto na seção 2.3 do Capítulo 2, mesmo as derivadas parciais de primeira ordem, representadas pelas equações (2.48) e (2.49), são expressões extensas e complexas, fazendo a obtenção e codificação de expressões de segunda ordem se tornarem um processo dispendioso e propenso a erros de implementação.

## 4.7 MÉTODO QUASI-NEWTON

Por fim, convém discutir o último método de geração da nova direção de busca  $\mathbf{p}_k$  nos algoritmos de otimização baseados no gradiente, o método Quasi-Newton.

Antes de propriamente derivar os resultados da família de métodos Quasi-Newton, se faz necessário citar suas vantagens em relação ao uso do método de Newton. Primeiramente, os métodos Quasi-Newton, assim como os métodos do gradiente máximo e do gradiente conjugado, necessitam apenas do gradiente da função objetivo  $f$  em cada iteração do algoritmo de otimização. Através da medição de mudanças no gradiente, os métodos Quasi-Newton constroem um modelo da função objetivo  $f$ , conforme será visto mais a frente, que é bom o suficiente para produzir uma nova geração de busca que possui rápida taxa de convergência [14]. Além do mais, como derivadas de segunda ordem não são necessárias, os métodos Quasi-Newton são, em certas ocasiões, mais eficientes do que os métodos de Newton [14]. Convém ainda notar que, o relativamente recente desenvolvimento de técnicas automáticas de diferenciação, tornaram possível o uso do método de Newton sem a implementação explícita de derivadas de segunda ordem da função objetivo  $f$ . Entretanto, as ferramentas de diferenciação automática ainda podem não ser aplicáveis em muitas situações práticas, além de ser muito mais complexo computacionalmente trabalhar com derivadas de segunda ordem em ferramentas de diferenciação automática do que com o gradiente apenas [14].

Em segundo lugar, pode-se citar que, devido à aproximação da matriz Hessiana  $\mathbf{H}$  nos métodos Quasi-Newton ser sempre positiva-definida, não se faz necessário criar modelos quadráticos modificados que garantam uma nova matriz Hessiana  $\mathbf{H}$  sempre positiva-definida, assim como utilizado nos métodos de Newton. O uso de técnicas como a adição de um múltiplo da matriz identidade e a resolução do sistema linear resultante, conforme o modelo quadrático modificado em (4.33), aumentam consideravelmente a complexidade de implementação computacional dos métodos de Newton.

Pelas razões citadas acima, escolheu-se o método Quasi-Newton como o algoritmo de otimização representativo da família de métodos de otimização baseado no gradiente, para resolução do problema de otimização no projeto de filtros digitais FIR de fase não-linear da seção 2.3 do Capítulo 2. Conforme será visto no Capítulo 7, esse algoritmo de otimização será extensivamente comparado, a nível de performance e qualidade da solução final  $\mathbf{x}^*$ , com outras abordagens de otimização baseadas em *soft-computing* apresentadas no Capítulo 5, aplicadas ao problema de otimização não-linear sem restrição.

Começa-se a derivação da família de métodos Quasi-Newton, pela formação do seguinte modelo quadrático  $m_k(\mathbf{p})$  da função objetivo pela expansão da série Taylor de segunda ordem no ponto  $\mathbf{x}_k$ :

$$m_k(\mathbf{p}) = f_k + \nabla f_k^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T \mathbf{B}_k \mathbf{p}, \quad (4.34)$$

onde  $\mathbf{B}_k$  é uma matriz positiva definida  $n \times n$  que será atualizada a cada iteração do algoritmo de otimização.

O minimizador  $\mathbf{p}_k$  do modelo convexo quadrático em (4.34), pode ser escrito explicitamente por [14]:

$$\mathbf{p}_k = -\mathbf{B}_k^{-1} \nabla f_k \quad (4.35)$$

E dessa forma, a relação em (4.35) é utilizada como a nova direção de busca no método Quasi-Newton.

Ao invés de recalcular  $\mathbf{B}_k$  à cada nova iteração, pode-se levar em consideração a curvatura medida durante o passo mais recente, obtendo um novo modelo quadrático da seguinte forma [14]:

$$m_{k+1}(\mathbf{p}) = f_{k+1} + \nabla f_{k+1}^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T \mathbf{B}_{k+1} \mathbf{p} \quad (4.36)$$

Um requisito razoável sobre esse novo modelo é que o gradiente de  $m_{k+1}$  deve coincidir com o gradiente da função objetivo  $f$  em pelo menos duas iterações anteriores  $\mathbf{x}_k$  e  $\mathbf{x}_{k+1}$ . Como  $\nabla m_{k+1}(0)$  é equivalente a  $\nabla f_{k+1}$ , o requisito sobre a iteração  $\mathbf{x}_{k+1}$  foi prontamente satisfeito. Com relação a iteração  $\mathbf{x}_k$ , pode-se escreve-la matematicamente como [14]:

$$\nabla m_{k+1}(-\alpha_k \mathbf{p}_k) = \nabla f_{k+1} - \alpha_k \mathbf{B}_{k+1} \mathbf{p}_k = \nabla f_k \quad (4.37)$$

Através de manipulação algébrica nos termos da condição em (4.37), em [14] obtém-se a seguinte fórmula:

$$\mathbf{B}_{k+1} \mathbf{s}_k = \mathbf{y}_k, \quad (4.38)$$

onde  $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k = \alpha_k \mathbf{p}_k$  e  $\mathbf{y}_k = \nabla f_{k+1} - \nabla f_k$ . A relação em (4.38) é conhecida como equação da secante. Essa equação exige que a matriz simétrica positiva-definida  $\mathbf{B}_{k+1}$  mapeie  $\mathbf{s}_k$  em  $\mathbf{y}_k$ . Isso é possível somente se  $\mathbf{s}_k$  e  $\mathbf{y}_k$  satisfaçam a condição de curvatura [14]:

$$\mathbf{s}_k^T \mathbf{y}_k > 0 \quad (4.39)$$

Quando a função objetivo  $f$  é convexa, a condição de curvatura em (4.39) será sempre satisfeita para quaisquer dois pontos  $\mathbf{x}_k$  e  $\mathbf{x}_{k+1}$  [14]. Entretanto, essa condição nem sempre se mantém para funções não-convexas, e nesse caso, a condição de curvatura em (4.39) deve ser imposta explicitamente, através de restrições no procedimento de busca em linha que escolhe o parâmetro  $\alpha$ . De fato a condição em (4.39) é garantida usando um método de busca em linha que atenda a condição de decrescimento suficiente em (4.12) e a condição de curvatura em (4.13), conforme o algoritmo apresentado detalhadamente na seção 4.3.

Quando a condição de curvatura é satisfeita, a equação da secante em (4.38) sempre tem uma solução  $\mathbf{B}_{k+1}$ . De fato, essa equação admite um número infinito de soluções, já que os  $\frac{n(n+1)}{2}$  graus de liberdade em uma matriz simétrica positiva-definida excedem as  $n$  condições impostas pela equação da secante [14]. Dessa forma, para se obter  $\mathbf{B}_{k+1}$  de forma única, impõe-se a condição adicional de que entre todas as matrizes simétricas positivas definidas que satisfazem a equação da secante,  $\mathbf{B}_{k+1}$  é, em alguma métrica (norma  $\ell_p$ ), mais próxima do que a matriz corrente  $\mathbf{B}_k$ . Ou seja, deve-se resolver o seguinte problema de otimização:

$$\min_{\mathbf{B}} \|\mathbf{B} - \mathbf{B}_k\| \quad \text{sujeito a} \quad \mathbf{B} = \mathbf{B}^T, \quad \mathbf{B}\mathbf{s}_k = \mathbf{y}_k, \quad (4.40)$$

e dessa forma, diferentes normas de matrizes podem ser usadas em (4.40), consequentemente resultando em diferentes métodos Quasi-Newton.

Antes de apresentar as diferentes formas de atualização da matriz  $\mathbf{B}_k$ , é oportuno apresentar a forma geral de um método Quasi-Newton, conforme definido no algoritmo 4.7.1.

---

**Algoritmo 4.7.1:** Estrutura geral de um algoritmo Quasi-Newton

---

- 4.7.1.1 Dado um ponto inicial  $\mathbf{x}_0$  e uma estimativa inicial da matriz Hessiana  $\mathbf{B}_0$ ;
- 4.7.1.2  $k \leftarrow 0$ ;
- 4.7.1.3 **enquanto** *Critério de tolerância da convergência do algoritmo for satisfeito* **faça**
- 4.7.1.4     Calcule a nova direção de busca  $\mathbf{p}_k \leftarrow -\mathbf{B}_k^{-1} \nabla f_k$ ;
- 4.7.1.5     Atribua  $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k \mathbf{p}_k$ , onde  $\alpha_k$  é obtido por um método de busca em linha;
- 4.7.1.6     Atualize a nova estimativa da matriz Hessiana  $\mathbf{B}_{k+1}$ ;
- 4.7.1.7      $k \leftarrow k + 1$ ;
- 

Duas importantes observações, em nível de implementação computacional, devem ser feitas com relação à forma geral de um algoritmo Quasi-Newton em 4.7.1. A primeira delas refere-se a qual deve ser a escolha da aproximação inicial da matriz Hessiana  $\mathbf{B}_0$ . Uma metodologia comum consiste em inicializar a estimativa com a matriz identidade,  $\mathbf{B}_0 = \mathbf{I}$ . Dessa forma, certamente  $\mathbf{B}_0$  é positiva-definida e essa inicialização resulta no primeiro passo dirigindo-se à direção do gradiente máximo [31]. Entretanto, essa metodologia possui a problemática de não considerar a escala da função objetivo  $f(\mathbf{x})$ . Por essa razão,  $\|\mathbf{B}_0\|$  pode ser diferente de  $\|\nabla^2 f(\mathbf{x}_0)\|$  por muitas ordens de magnitude [31]. Entre as várias estratégias propostas na literatura para superar esse problema, pode-se citar a estratégia que utiliza um caso especial da idéia mais geral de pré-multiplicar  $\mathbf{B}_0$  por  $\gamma_0 = \frac{\mathbf{y}_0^T \mathbf{s}_0}{\mathbf{s}_0^T \mathbf{B}_0 \mathbf{s}_0}$  imediatamente antes da primeira atualização. Ainda que essa metodologia apresente bons resultados, devido a alguns pequenos problemas adicionais [31], a implementação detalhada do algoritmo Quasi-Newton desenvolvida no algoritmo I.1.1 do apêndice I utiliza a metodologia desenvolvida em [31], a qual foi reportada com base em resultado numéricos experimentais, como a melhor abordagem de inicialização da estimativa da matriz Hessiana  $\mathbf{B}_0$ . A seção I.1.1 no apêndice I apresenta de forma detalhada essa abordagem de inicialização.



A segunda observação contempla a questão de como melhor representar a aproximação da matriz Hessiana. Uma metodologia utilizada logo no surgimento dos métodos Quasi-Newton consiste em vez de se armazenar a forma direta da aproximação de  $\nabla^2 f_k$ , guardar a aproximação da matriz inversa da Hessiana  $\nabla^2 f_k^{-1}$ , já que essa forma de representação evita a resolução do sistema de equações [10]:

$$\mathbf{B}_k \mathbf{p}_k = -\nabla f_k \quad (4.41)$$

Outra abordagem consiste em armazenar a forma direta da aproximação da Hessiana  $\mathbf{H}_k$ . Com essa abordagem, duas formas de implementação surgem, a forma não-fatorada e a forma fatorada. Na primeira forma, simplesmente implementa-se alguma das fórmulas de atualização da aproximação da matriz Hessiana, conforme serão definidas a seguir. Já na segunda forma, particiona-se a estimativa da Hessiana em duas matrizes triangulares através da operação de fatorização Cholesky e da atualização das matrizes fatoradas [10]. Em termos de complexidade computacional, a forma fatorada realiza  $O(n^2)$  operações, enquanto que a forma não-fatorada  $O(n^3)$  operações. Assim, embora a forma não-fatorada possua maior custo computacional, ela é especialmente mais fácil de ser programada. Devido a esse fator, optou-se nesse trabalho pelo uso da forma não-fatorada, já que a medida comparativa de performance entre os vários algoritmos de otimização realizadas no Capítulo 7 não será a complexidade computacional, mas sim, o número de avaliações da função objetivo  $f$ . Isso se deve ao fato da dificuldade de geração de modelos matemáticos e de ferramentas que possibilitem analisar detalhadamente o comportamento dos algoritmos do Capítulo 5. Embora algumas técnicas tenham sido desenvolvidas na literatura, como por exemplo em [32], estudos mais aprofundados necessitam ser conduzidos para analisar os algoritmos baseados em *soft-computing* e generalizar seus comportamentos em problemas práticos de maior complexidade [32]. Dessa forma, nenhum ganho de performance é obtido no comparativo pelo uso da representação fatorada nessa situação em particular.

Tendo sido feito o comparativo entre o método Quasi-Newton e o método de Newton, derivado a teoria da equação da secante com o modelo quadrático modificado e apresentado a forma genérica da família de algoritmos Quasi-Newton, além de abordar algumas questões de implementação, se faz necessário agora apresentar as diferentes fórmulas de atualização da estimativa da matriz Hessiana  $\mathbf{B}_k$ . Conforme visto a partir do problema de minimização da distância de aproximação da matriz Hessiana na fórmula da secante em (4.40), diferentes funções de aproximação correspondem a diferentes métodos Quasi-Newton. Nesse trabalho, não será apresentada a derivação das diferentes fórmulas de atualização da estimativa da matriz Hessiana  $\mathbf{B}_k$ , já que as derivações são relativamente extensas e fogem do escopo do presente trabalho. A derivação completa das fórmulas encontram-se nas referências [14], [10], [31] ou [17].

Cronologicamente, o primeiro método de atualização da estimativa da matriz Hessiana foi creditado aos seus respectivos criadores, Davidon-Fletcher-Powell (DFP). Sua forma de atualização é dada pela seguinte relação [31]:



$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{(\mathbf{y}_k - \mathbf{B}_k \mathbf{s}_k) \mathbf{y}_k^T + \mathbf{y}_k (\mathbf{y}_k - \mathbf{H}_k \mathbf{s}_k)^T}{\mathbf{y}_k^T \mathbf{s}_k} - \frac{\langle \mathbf{y}_k - \mathbf{B}_k \mathbf{s}_k, \mathbf{s}_k \rangle \mathbf{y}_k \mathbf{y}_k^T}{(\mathbf{y}_k^T \mathbf{s}_k)^2}, \quad (4.42)$$

onde o operador  $\langle \mathbf{a}, \mathbf{b} \rangle$  denota o produto interno entre os vetores  $\mathbf{a}$  e  $\mathbf{b}$ . Em [31], mostra-se que se a condição de curvatura  $\mathbf{y}_k^T \mathbf{s}_k > 0$  é mantida e a matriz  $\mathbf{B}_k$  é positiva-definida, então a atualização DFP em (4.42) também gera uma atualização  $\mathbf{B}_{k+1}$  positiva-definida.

O outro método de atualização que será apresentado, e talvez o mais utilizado da família de métodos Quasi-Newton consiste no método BFGS, nome dado aos seus respectivos criadores Broyden-Fletcher-Goldfarb-Shanno. Sua forma de atualização da estimativa da matriz Hessiana  $\mathbf{B}_{k+1}$  é dada pela seguinte relação [31]:

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k} - \frac{\mathbf{B}_k \mathbf{s}_k \mathbf{s}_k^T \mathbf{B}_k}{\mathbf{s}_k^T \mathbf{B}_k \mathbf{s}_k} \quad (4.43)$$

A fórmula DFP em (4.42) é bastante efetiva, mas ela foi rapidamente substituída pelo método BFGS em (4.43), o qual é atualmente considerado o método mais efetivo de todos os métodos existentes da família de algoritmos Quasi-Newton [14]. Devido à essa observação, no presente trabalho usa-se exclusivamente o método BFGS na implementação do algoritmo de otimização Quasi-Newton.

Por fim, faz-se lembrar que alguns detalhes de implementação não citados nesse capítulo, assim como a definição formal dos passos do algoritmo Quasi-Newton em conjunto com o método de busca em linha que obedece as condições de Wolfe se encontra na seção I.1 do apêndice I.

# 5 ALGORITMOS DE OTIMIZAÇÃO SOFT-COMPUTING

## 5.1 INTRODUÇÃO

A partir de origem completamente diversa das técnicas de otimização definidas no Capítulo 4, novos paradigmas computacionais com origem na área de inteligência artificial vem sendo amplamente estudados.

Um desses paradigmas, *soft-computing*, termo criado pelo prof. Zadeh em 1992 após o surgimento da área de inteligência computacional no começo dos anos 90, é definido por quatro elementos fundamentais [33]:

- Lógica dos conjuntos nebulosos.
- Modelos computacionais de redes neuronais.
- Computação evolucionária.
- Redes Bayesianas.

Embora a área de inteligência computacional englobe os elementos definidos acima, o termo *soft-computing* geralmente é utilizado de forma mais ampla, já que o mesmo visa a sinergia dessas ferramentas computacionais. Conforme as palavras de Zadeh, uma de suas definições para *soft-computing* é [34]:

*Soft-computing* não é um conjunto homogêneo de conceitos e técnicas. Ao invés disso, é uma coleção de metodologias, as quais de uma forma ou de outra refletem os princípios orientadores da *soft-computing*: explorar a tolerância para imprecisão, incerteza e crença parcial para conseguir-se tratabilidade, robustez e soluções de baixo custo. Visto de uma perspectiva ligeiramente diferente, *soft-computing* é um consórcio de metodologias, as quais, sendo de forma única ou combinada, servem para oferecer ferramentas efetivas para o desenvolvimento de sistemas inteligentes.

Os algoritmos *soft-computing* apresentados nesse capítulo, também classificados de algoritmos de otimização estocásticos, possuem certas vantagens em relação aos métodos de otimização baseados em gradientes explanados no Capítulo 4. Enquanto os algoritmos daquele capítulo apresentam melhores resultados e melhor eficiência computacional em problemas lineares, quadráticos, convexos, unimodais e em outras aplicações especializadas, os algoritmos *soft-computing*

vêm se mostrando mais eficientes para problemas descontínuos, não-diferenciáveis, multimodais e ruidosos [16].

Conforme visto no Capítulo 3, especificamente no caso de projeto de filtros FIR de fase não-linear, a função objetivo para otimização (2.47) é não-linear e multimodal, possuindo assim, um ambiente propício ao uso de algoritmos *soft-computing* para esse problema particular de otimização não-linear sem restrição. Os resultados apresentados no Capítulo 7 confirmam essas afirmações.

Entretanto, uma problemática encontrada no uso desses algoritmos se deve a seleção de seus parâmetros. Exemplos desses parâmetros incluem o tamanho da população, taxas de mutação e *crossover*, tamanho dos grupos elitistas, constantes de aceleração, tamanho de passos ... Valores para esses parâmetros são comumente ajustados através de um processo de tentativa e erro, em vez de uma sólida análise estatística de seus efeitos na performance algorítmica [35]. Tendo em vista essa falta de rigor e metodologia na escolha dos parâmetros, o Capítulo 6 apresenta uma metodologia de análise exploratória estatística dos parâmetros dos algoritmos *soft-computing* o qual evita o uso desses algoritmos com parâmetros sub-ótimos.

O restante desse capítulo é organizado da seguinte forma. A seção 5.2 apresenta algumas das técnicas disponíveis para inicialização da população nos algoritmos evolucionários. A seção 5.3 define a função de adequação, uma medida de sobrevivência dos indivíduos no contexto dos algoritmos evolucionários, ou a função de energia do sistema no contexto do resfriamento simulado. Por sua vez, a seção 5.4 apresenta o provavelmente mais conhecido algoritmo de otimização evolucionário, o algoritmo genético, que simula o processo natural de evolução das espécies através de uma busca paralela estocástica pelo espaço de soluções. A seção 5.5 apresenta uma técnica de otimização estocástica baseada na processo físico de resfriamento de um líquido ou sólido e como esse sistema tende a alcançar seu estado mínimo de energia, o resfriamento simulado. A seção 5.6 apresenta uma técnica relativamente recente de otimização evolucionária, a otimização por enxame de partículas, a qual simula a busca cooperativa de partículas no espaço de soluções.

## 5.2 INICIALIZAÇÃO DA POPULAÇÃO

Devido a sua natureza estocástica e ao processo inerente de busca baseado em população nos algoritmos evolucionários, cada algoritmo mantém uma população de soluções candidatas. Dessa forma, o primeiro passo para uso desses algoritmos consiste na inicialização de sua população.

A tarefa de inicialização da população é crucial em algoritmos evolucionários já que ela pode afetar a velocidade de convergência e também a qualidade da solução final [36]. Se não é disponível *a priori* nenhum conhecimento sobre a solução, então, uma metodologia comum consiste na inicialização aleatória, com determinada distribuição de probabilidade, dos indivíduos da população. Embora o problema de otimização de filtros digitais definido no Capítulo 2 possa se

beneficiar de uma estimativa inicial da solução ótima, para efeitos comparativos, conforme será visto no Capítulo 7, e para medição do poder de otimização dos algoritmos sem uso de nenhuma heurística da estrutura do problema, optou-se pela inicialização da população sem nenhum conhecimento *a priori* do espaço de solução.

Para a inicialização aleatória dos indivíduos da população, alguns melhoramentos que afetam a velocidade de convergência e a qualidade final da solução foram propostos, entre eles, optou-se na implementação dos algoritmos evolucionários por uma técnica de inicialização com aprendizado baseado em oposição [36]. Nesse esquema, a idéia principal é a consideração de estimativas e estimativas opostas ao mesmo tempo, para dessa forma, conseguir-se uma melhor aproximação das soluções candidatas correntes. O processo de inicialização da população proposto no presente trabalho é definido no algoritmo 5.2.1 [36]. Nesse algoritmo,  $n$  representa o tamanho da população e o operador de oposição  $O(\cdot)$  é definido por:

$$O(P_{k,j}) = a_j + b_j - P_{k,j}, \quad k = 1, 2, \dots, n \quad j = 1, 2, \dots, N_v \quad (5.1)$$

---

**Algoritmo 5.2.1:** Inicialização da população com aprendizado baseado em oposição

---

- 5.2.1.1 Gerar população aleatória  $P(n)$  distribuída uniformemente;
  - 5.2.1.2 Calcular população oposta  $O(P(n))$ . O  $k$ -ésimo indivíduo oposto correspondente para  $O(P(n))$  é definido na equação (5.1);
  - 5.2.1.3 Selecionar  $n$  indivíduos de maior função de adequação do conjunto  $P(n) \cup O(P(n))$  como população inicial;
- 

, onde na equação (5.1),  $N_v$  é o número de variáveis (dimensão do problema),  $a_j$  e  $b_j$  denotam as fronteiras do intervalo da  $j$ -ésima variável ( $x_j \in [a_j, b_j]$ ).

Conforme será visto no Capítulo 7, a métrica para comparação de performance entre os algoritmos de otimização é o número de avaliações da função objetivo  $N_{av}$ , ou função de adequação no caso dos algoritmos evolucionários. Embora no algoritmo de inicialização 5.2.1  $N_{av}$  seja o dobro,  $N_{av} = n$ , do que em um simples algoritmo de inicialização populacional aleatório,  $N_{av} = \frac{n}{2}$ , considerando a otimização como um todo, o algoritmo 5.2.1 possuiu melhor performance em quase todos testes de otimização em [36]. Isso se deve ao algoritmo de otimização ser inicializado com indivíduos mais aptos, sendo que na média, mesmo que se percam algumas avaliações funcionais na inicialização, ganha-se em performance na otimização como um todo, resultando em maior velocidade de convergência.

## 5.3 FUNÇÃO DE ADEQUAÇÃO

No modelo Darwinista de evolução, indivíduos com as melhores características tem melhores chances de sobreviver e se reproduzir. Para determinar a habilidade de sobrevivência de um indi-

víduo em um algoritmo evolucionário, uma função matemática é utilizada para quantificar o quão bom é a solução representada pelo indivíduo do algoritmo evolucionário [16]. Essa função é chamada de função de adequação e mapeia a representação do indivíduo no algoritmo evolucionário em um valor escalar:

$$f: \mathbf{X}^{n_x} \rightarrow \mathbb{R}, \quad (5.2)$$

onde  $\mathbf{X}$  representa o indivíduo codificado no algoritmo evolucionário e  $n_x$  é a dimensão do espaço de busca do indivíduo.

Conforme visto no Capítulo 3, vários tipos de problemas de otimização foram apresentados. Dessa forma, é importante notar que esses problemas de otimização possuem influência na formulação da função de adequação. Particularmente, para o problema do projeto de filtros FIR com fase não-linear com função objetivo dado pela equação (2.47), um problema multi-objetivo sem restrição, a função de adequação é mapeada diretamente na função objetivo para um problema de maximização ou como uma função inversa para o problema de minimização. Além disso, devido à natureza multi-objetivo da função em (2.47), essa possui a forma de uma soma ponderada das sub-funções objetivos, conforme visto no Capítulo 3. Logo, a função de adequação dos algoritmos evolucionários implementados nesse trabalho possui a seguinte forma:

$$f(\mathbf{X}) = \frac{1}{1 + L_{2p,2q}^{\alpha,\tau}(\Phi(\mathbf{X}))}, \quad (5.3)$$

onde  $\Phi(\mathbf{X})$  representa genericamente uma função de decodificação da representação do indivíduo no algoritmo evolucionário para o vetor  $\mathbf{A}$  de zeros  $z_i$  complexos conjugados da função de transferência em (2.36):

$$\Phi: \mathbf{X} \mapsto \mathbf{A} \quad (5.4)$$

Convém notar que a função de adequação em (5.3) possui contra-domínio entre  $0 \leq f(\mathbf{X}) \leq 1$ , assim:

$$f: \mathbb{C} \rightarrow \mathbb{R}_0^1 \quad (5.5)$$

Particularmente para o algoritmo resfriamento simulado, conforme será visto na seção 5.5, diferentemente dos algoritmos evolucionários, não existe exatamente uma função de adequação, mas sim, uma função de energia  $E(\mathbf{X})$  do sistema. Nesse caso é feito o mapeamento direto da função objetivo em (2.47) para a função de energia  $E(\mathbf{X})$ :

$$E(\mathbf{X}) = L_{2p,2q}^{\alpha,\tau}(\Phi(\mathbf{X})), \quad (5.6)$$

onde nesse caso,  $\mathbf{X}$  representa uma configuração particular dos átomos no algoritmo resfriamento simulado e  $\Phi(\mathbf{X})$ , equação (5.4), representa a função de decodificação da configuração dos átomos no algoritmo para o vetor  $\mathbf{A}$  de zeros  $z_i$  complexos conjugados da função de transferência em (2.36).

## 5.4 ALGORITMOS GENÉTICOS

Os algoritmos genéticos foram inventados por John Holland nos anos 60, e desenvolvidos por Holland e seus estudantes e colegas na Universidade de Michigan nos anos 60 e 70 [37]. O objetivo original de Holland com seu algoritmo genético foi o de estudar formalmente o fenômeno de adaptação que ocorre na natureza e de desenvolver maneiras nas quais o mecanismo de adaptação natural possa ser incorporados em sistemas computacionais.

O algoritmo genético é um método para mover-se de uma população de cromossomos para uma nova população através de uma espécie de seleção natural junto com operadores inspirados na genética, *crossover*, mutação e inversão [37]. Cada cromossomo consiste de genes, ex: bits ou número de ponto flutuante. O operador de seleção escolhe aqueles cromossomos na população que terão permissão para reproduzir, e na média, o cromossomo mais apto produz mais descendentes do que os menos aptos. Já o operador de *crossover* troca sub-partes de dois cromossomos, simplificadaamente simulando a recombinação biológica entre dois cromossomos. Por fim o operador mutação muda aleatoriamente os valores dos alelos, valor particular de um gene, em algumas posições do cromossomo.

Outra interpretação para o fenômeno de adaptação ou evolução que ocorre na natureza é a de um processo de busca através do espaço de todos os possíveis valores dos cromossomos. Nesse sentido, o algoritmo genético executa uma busca estocástica para uma solução ótima para um dado problema. O processo de evolução nos algoritmos genéticos são influenciados pelos seguintes componentes principais [16]:

- Codificação das soluções para o problema através de cromossomos.
- Função para avaliar a adequação ou força de sobrevivência dos indivíduos.
- Inicialização da população.
- Operador de seleção.
- Operadores de reprodução.

A inicialização da população e a função de adequação foram discutidas no contexto mais geral dos algoritmos evolucionários respectivamente nas seções 5.2 e 5.3. O que se segue nessa seção é a discussão detalhada dos componentes restantes, respectivamente nos itens 5.4.1, 5.4.2 e 5.4.3.

Por fim, convém notar que a descrição detalhada do algoritmo genético com a implementação da codificação das soluções, inicialização da população, seleção e reprodução em forma algorítmica se encontra na subseção I.2.1 do apêndice I.

#### 5.4.1 Codificação das soluções candidatas

Na natureza, os organismos possuem certas características que influenciam suas habilidades para sobrevivência e reprodução. Essas características são representadas por longas cadeias de informação contidas nos cromossomos dos organismos. Cada cromossomo contém um grande número de genes, onde cada gene é a unidade de hereditariedade. Os genes determinam muitos dos aspectos anatômicos e fisiológicos através do controle de produção de proteínas. Cada indivíduo possui uma sequência única de genes [16].

No contexto dos algoritmos genéticos, cada indivíduo representa uma solução candidata para um problema de otimização. As características do indivíduo são representadas por um cromossomo e tais características referem-se às variáveis do problema de otimização.

Um aspecto crucial nos algoritmos genéticos consiste em achar uma representação apropriada das soluções candidatas. A eficiência e complexidade do algoritmo de busca dependem em grande parte do esquema de representação [16]. Esquemas de codificação das soluções candidatas podem ser genericamente classificados em duas categorias:

- Representações discretas com códigos binários.
- Representações contínuas com números de ponto flutuante.

Na representação com códigos binários, considerando um espaço de busca de dimensão  $N$ , cada indivíduo consistirá de  $N$  variáveis, com cada variável codificada como um conjunto de  $n$  bits. Com a representação binária, cada variável codificada poderá ter  $2^n$  valores discretos representados. Para problemas de otimização com variáveis de valores contínuos, necessita-se de um mapeamento do espaço binário  $\{0, 1\}^N$  para o espaço contínuo  $\mathbb{R}^N$ . Vários esquemas podem ser delineados para esse mapeamento como o uso direto de código binário para representação de números contínuos (ponto-flutuante), o uso de código Gray e outros mapeamentos menos convencionais, que em determinados domínios de aplicação possuem melhor performance, como códigos uniformes, códigos tri-setores e códigos simétricos [38].

Já na representação com números de ponto flutuante, novamente considerando um espaço de busca de dimensão  $N$ , cada indivíduo consistirá de  $N$  variáveis, com cada variável codificada como um número de ponto flutuante. Essa forma de representação é robusta, precisa e eficiente já que a representação de ponto flutuante é conceitualmente mais próxima ao espaço de busca real do problema de otimização [39]. Em várias aplicações reais, vem se reportando na literatura melhor performance da codificação com número de ponto flutuante em vez da codificação com códigos binários [39].

Especificamente nesse trabalho, para otimização da função objetivo em (2.47), devido à estrutura da função transferência na forma pólo-zero-ganho (2.36) do filtro FIR de fase não-linear, cada variável do algoritmo genético é representada por um  $i$ -ésimo zero  $z_{zi}$  da função de transferência (2.36). Convém lembrar que tais zeros são definidos no domínio complexo  $z_{zi} \in \mathbb{C}$  e aparecem na função transferência (2.36) em pares complexos conjugados. Dessa forma, na implementação do algoritmo genético para resolução do problema de aproximação no projeto de filtros FIR de fase não-linear, cada variável é representada por dois números de ponto flutuante, respectivamente o módulo  $|z_{zi}|$  e fase  $\angle z_{zi}$  dos zeros da função de transferência em coordenadas polares (2.38). Além disso, para cada  $i$ -ésimo zero  $z_{zi}$  da função de transferência é mantido internamente no algoritmo seu  $i$ -ésimo complexo conjugado  $z_{zi}^*$ , já que para obtenção dos coeficientes reais  $b_i$  do filtro no final da otimização, deve-se expandir o produto da função de transferência na forma pólo-zero-ganho em (2.36) para obtenção de uma função de transferência direta na forma (2.8) com os  $b_i$  coeficientes reais do filtro.

### 5.4.2 Operador de Seleção

O processo de seleção se relaciona diretamente com o conceito Darwinista de sobrevivência do mais apto. O principal objetivo do operador de seleção é de enfatizar melhores soluções [16]. De forma genérica, uma nova população de soluções candidatas é selecionada no fim de cada geração para servir como população da próxima geração. A nova população pode ser selecionada apenas pelos descendentes ou através dos pais e de seus descendentes. Dessa forma, o operador de seleção deve garantir que bons indivíduos sobrevivam nas próximas gerações [16].

Na literatura de algoritmos genéticos, uma diversidade de operadores de seleção foram criados. Entre eles podem-se citar:

- Seleção aleatória.
- Seleção proporcional.
- Seleção por torneio.
- Seleção baseada em *rank*.
- Seleção Boltzmann.
- Elitismo.

Em [40] resultados comparativos da complexidade em função do tempo e da taxa de crescimento da população são reportados. A partir dos resultados, é bastante improvável que um método de seleção particular seja o melhor para os todos problemas de otimização. Nesse trabalho, optou-se pela implementação do algoritmo genético com o método de seleção proporcional, principalmente pela sua simplicidade e pelo extenso uso na literatura.



### 5.4.2.1 Seleção proporcional

Nesse método, o processo de seleção tende a escolher os indivíduos mais aptos da população. Uma distribuição de probabilidade proporcional à função de adequação (5.7) é criada e os indivíduos são selecionados por amostragem dessa distribuição [16].

$$\varphi_s(\mathbf{x}_i(t)) = \frac{f(\mathbf{x}_i(t))}{\sum_{l=1}^{n_s} f(\mathbf{x}_l(t))}, \quad (5.7)$$

onde  $n_s$  é o número total de indivíduos da população,  $\mathbf{x}_i$  é o cromossomo, vetor de pontos flutuantes definidos no item 5.4.1,  $\varphi_s(\mathbf{x}_i(t))$  é a probabilidade que  $\mathbf{x}_i$  seja selecionado e  $f(\mathbf{x}_i(t))$  é a função de adequação de  $\mathbf{x}_i$ .

Um algoritmo popular que implementa seleção proporcional é o *Roulette Wheel*, apresentado no algoritmo 5.4.1 [16].

---

**Algoritmo 5.4.1: Roulette Wheel**

---

5.4.1.1 Atribuir  $i = 1$ , onde  $i$  denota o índice do cromossomo ;

5.4.1.2 Calcular  $\varphi_s(\mathbf{x}_i(t))$  ;

5.4.1.3 **somatorio** =  $\varphi_s(\mathbf{x}_i(t))$ ;

5.4.1.4 Escolher  $r \sim U(0, 1)$ ;

5.4.1.5 **enquanto** somatorio <  $r$  **faça**

5.4.1.6      $i = i + 1$ ;

5.4.1.7     **somatorio** = somatorio +  $\varphi_s(\mathbf{x}_i(t))$ ;

5.4.1.8 Retornar  $\mathbf{x}_i$  como indivíduo selecionado;

---

### 5.4.3 Operadores de Reprodução

De forma genérica, reprodução é o processo de produção de novas gerações de pais selecionados, aplicando-se operadores de *crossover* e mutação [16].

No operador de *crossover* criam-se um ou mais indivíduos através da combinação de material genético selecionado aleatoriamente de dois ou mais pais. Já no operador de mutação, mudanças aleatórias são feitas nos valores dos genes. Dessa forma, o objetivo principal desse operador é o de introduzir novo material genético na população, aumentando a diversidade genética.

Embora no algoritmo genético clássico, os operadores de reprodução definidos sejam os de *crossover* e mutação, vários outros operadores, com aplicações especializadas existem na literatura, como por exemplo, o uso de operadores de reprodução com informação do gradiente da função objetivo [41], para aplicação em algoritmos com custo computacional e número reduzido de indivíduos.

Nesse trabalho, optou-se pela implementação clássica do algoritmo genético com os operadores de *crossover* e mutação, visando a simplicidade e tratabilidade da análise estatística dos

parâmetros dos algoritmos evolucionários, conforme será desenvolvida no Capítulo 6.

#### 5.4.3.1 Operador de crossover

Na literatura dos algoritmos genéticos, acredita-se que o operador de *crossover* seja o principal operador de busca no processo de otimização do algoritmo [42]. O operador de *crossover* possui basicamente o propósito de combinar, após a seleção dos cromossomos aleatórios iniciais mais aptos, porções desses cromossomos para formação de indivíduos ainda mais aptos.

O sucesso do operador de *crossover* é largamente dependente do esquema de codificação utilizado para as variáveis do problema. Caso as variáveis do problema de otimização não estejam codificadas apropriadamente, o operador de *crossover* não pode combinar corretamente os cromossomos dos indivíduos da população [42].

Conforme visto no item 5.4.1, para resolução de problemas com espaço de busca contínuo, uma das alternativas de codificação consiste no uso de códigos binários para codificação das variáveis do problema. Entretanto, essa alternativa de codificação apresenta uma série de problemas, entre eles, a dificuldade para conseguir precisão arbitrária das variáveis para uma solução ótima do problema de otimização [42]. Motivado pelo sucesso dos algoritmos genéticos com codificação binária e pelo poder de busca do operador de *crossover* da versão binária do algoritmo, em [42] é derivado o operador de *crossover* denominado por *Simulated Binary Crossover* (SBX), o qual dispensa o uso de codificação binária das variáveis do problema e ainda assim, possui poder de busca comparável à sua versão binária.

O operador SBX é um operador que simula o *crossover* de um único ponto. Ou seja, nesse operador, um ponto de *crossover* é selecionado aleatoriamente e os genes após aquele ponto são trocados entre os dois pais [16]. Dessa forma, os pais  $x_1(t)$  e  $x_2(t)$  são usados para produzir dois descendentes  $\tilde{x}_1(t)$  e  $\tilde{x}_2(t)$  a partir das seguintes relações [16]:

$$\begin{aligned}\tilde{x}_{1j}(t) &= 0.5 [(1 + \gamma_j) x_{1j}(t) + (1 - \gamma_j) x_{2j}(t)] \\ \tilde{x}_{2j}(t) &= 0.5 [(1 - \gamma_j) x_{1j}(t) + (1 + \gamma_j) x_{2j}(t)],\end{aligned}\tag{5.8}$$

onde  $j = 1, \dots, n_x$ ,  $n_x$  é o número de indivíduos na população e:

$$\gamma_j = \begin{cases} (2r_j)^{\frac{1}{\eta+1}} & \text{se } r_j \leq 0.5 \\ \left(\frac{1}{2(1-r_j)}\right)^{\frac{1}{\eta+1}} & \text{caso contrário} \end{cases},\tag{5.9}$$

com  $r_j \sim U(0, 1)$  e  $\eta$  é o índice da distribuição. Em [42], sugere-se  $\eta = 1$ .

Convém lembrar que o operador de *crossover* é aplicado com probabilidade  $p_{cr}$  em cada par de pais da população. Conforme será visto no Capítulo 6, para ajuste dessa taxa de *crossover*  $p_{cr}$

será aplicada uma metodologia exploratória de análise estatística para obtenção de valores ótimos que levam à melhor performance de otimização.

O poder de busca do operador de *crossover* pode ser definido como uma medida de quão flexível o operador é para criar um ponto arbitrário no espaço de busca. Em [42] deriva-se o poder de busca do operador de *crossover* binário e expressões com poder de busca similar são derivadas para o operador SBX.

#### 5.4.3.2 Operador de mutação

O operador mutação tem por objetivo introduzir novo material genético a um indivíduo existente, ou seja, adicionar diversidade às características genéticas da população [16]. Da mesma forma que o *crossover*, o operador de mutação é aplicado com probabilidade  $p_{mu}$  a cada gene do  $i$ -ésimo cromossomo descendente  $\tilde{x}_i(t)$  para produzir um descendente mutante  $x'_i(t)$ . Novamente, para ajuste dessa taxa de mutação  $p_{mu}$ , no Capítulo 6 será aplicada uma metodologia exploratória de análise estatística para obtenção de valores ótimos que levam à melhor performance de otimização.

O operador de mutação empregado nesse trabalho para variáveis contínuas segue a metodologia em [43], uma das primeira propostas do operador sobre números contínuos [16]. Dessa forma, o operador de mutação é definido pelo seguinte par de equações [43]:

$$x'_i(t) = \begin{cases} \tilde{x}_i(t) + \Delta(t, UB - \tilde{x}_i(t)) & \text{se } r_b = 0 \\ \tilde{x}_i(t) - \Delta(t, \tilde{x}_i(t) - LB) & \text{se } r_b = 1 \end{cases}, \quad (5.10)$$

onde  $r_b$  é um bit aleatório com distribuição uniforme,  $LB$  e  $UB$  são respectivamente os limites inferiores e superiores da variável  $\tilde{x}_i(t)$  e  $\Delta(t, y)$  é uma função de distribuição não-uniforme definida por [43]:

$$\Delta(t, y) = y \cdot \left( 1 - r^{(1 - \frac{t_{av}}{T_{av}})^b} \right), \quad (5.11)$$

onde  $r \sim U(0, 1)$ ,  $T_{av}$  é o número máximo de avaliações da função de custo definido na otimização, conforme será visto no Capítulo 7,  $t_{av}$  é o número de avaliações atual da função de custo e  $b$  é um parâmetro do sistema. Em [43] escolheu-se  $b = 5$  assim como no presente trabalho. Nota-se que pelo uso de  $\Delta(t, y)$  o operador de mutação possui a propriedade inicial de busca uniforme do espaço de soluções, e na medida que  $t_{av}$  é incrementado o operador se degenera à uma busca local da solução [43].

## 5.5 RESFRIAMENTO SIMULADO

O desenvolvimento do algoritmo resfriamento simulado (RS), foi motivado por estudos em mecânica estatística, a qual lida com o equilíbrio de um grande número de átomos em sólidos e líquidos numa dada temperatura. Para conseguir atingir um estado de energia mínimo global e estável, o processo de resfriamento, o mesmo que se aplica em metalurgia é utilizado, onde nesse processo, o metal é re-aquecido à uma alta temperatura e depois resfriado lentamente, permitindo aos átomos tempo suficiente para achar posições que minimizem um estado de energia potencial estável [44].

O RS é um algoritmo de otimização baseado no processo físico descrito acima. No contexto da teoria de otimização, o mínimo de uma função objetivo representa a energia mínima do sistema. Dessa forma, o RS é uma implementação algorítmica de um processo de resfriamento para achar o valor ótimo da função objetivo [16].

Um aspecto único do RS se deve ao fato de que o mesmo utiliza uma estratégia de busca aleatória que aceita não somente posições que diminuem a função objetivo (mas também que aumentam), considerando um problema de minimização.

Basicamente o RS é composto de 3 operações: geração de vizinhança com uma função densidade de probabilidade  $g$ , transição de estado com uma probabilidade de aceitação  $h$  e diminuição da temperatura com um fator de temperatura  $T$  [45].

Uma propriedade desejada do RS consiste num esquema rápido de resfriamento, ex: convergência média rápida ao mínimo global. Isso significa que o algoritmo deve degenerar para uma busca aleatória, condição necessária para convergência [46]. O esquema clássico de resfriamento, também chamada de resfriamento Boltzmann, foi primeiramente empregado como uma técnica de amostragem de Monte Carlo para cálculo de integrais de linhas de alta dimensionalidade originárias de problemas em física estatística [47]. Logo após, esse método foi generalizado para aplicações mais abrangentes de otimização de funções não-convexas de problemas diversos [48], tendo sido aplicado pioneiramente em problemas discretos de alocação ótima de conexões em chips de computadores [49]. Embora vários esquemas de resfriamento existam na literatura, [48], [46] e [45], optou-se pelo resfriamento Boltzmann devido à sua simplicidade de implementação e amplo uso na literatura.

Antes de discutir detalhadamente o resfriamento Boltzmann, convém notar que a descrição detalhada do algoritmo RS com a implementação algorítmica da simulação do processo físico de resfriamento que minimiza um estado de energia potencial estável, com a finalidade de minimização da função objetivo 2.47 do projeto de filtros FIR de fase não-linear, se encontra na subseção I.2.2 do apêndice I.

### 5.5.1 Resfriamento Boltzmann

Conforme discutido acima, o RS é composto de 3 operações: geração de vizinhança com uma função densidade de probabilidade  $g$ , transição de estado com uma probabilidade de aceitação  $h$  e diminuição da temperatura com um fator de temperatura  $T$ . Esses elementos são definidos para o resfriamento Boltzmann por:

- Densidade de probabilidade de geração da vizinhança  $g(x)$ , equação (5.12). É baseada na forma funcional derivada de muitos sistemas físicos pertencentes à classe de sistemas Gaussianos-Markovianos [48].

$$g(x) = (2\pi T)^{-\frac{D}{2}} e^{-\frac{\Delta x^2}{2T}}, \quad (5.12)$$

onde  $D$  é a dimensão do espaço de busca,  $x$  é um ponto no espaço de busca,  $\Delta x = x - x_0$  é o desvio do ponto  $x$  para  $x_0$  e  $T$  é uma medida da flutuação da distribuição de Boltzmann  $g$  no espaço de busca.

- Densidade de probabilidade para aceitação da nova função objetivo  $h(x)$ , equação (5.13), dado o valor imediatamente anterior.

$$h(x) = \frac{1}{1 + e^{\frac{\Delta E}{T}}}, \quad (5.13)$$

onde  $\Delta E$  representa a diferença energética entre o valor presente e anterior da função objetivo.

- Diminuição da temperatura  $T$ , equação (5.14).

$$T(k) = \frac{T_0}{\ln k}, \quad (5.14)$$

onde  $k$  é o índice temporal discreto da simulação.

A função de energia  $E$  do RS foi definida em (5.6). Cada  $i$ -ésima posição  $x$  no espaço de busca do RS é convertida para um  $i$ -ésimo zero  $z_{zi}$  da função de transferência (2.36). Novamente, convém lembrar que tais zeros são definidos no domínio complexo  $z_{zi} \in \mathbb{C}$  e aparecem na função transferência (2.36) em pares complexos conjugados. Dessa forma, na implementação do RS para resolução do problema de aproximação no projeto de filtros FIR de fase não-linear, cada variável é representada por dois números de ponto flutuante, respectivamente o módulo  $|z_{zi}|$  e fase  $\angle z_{zi}$  dos zeros da função de transferência em coordenadas polares (2.38). Além disso, para cada  $i$ -ésimo zero  $z_{zi}$  da função de transferência é mantido internamente no algoritmo seu  $i$ -ésimo complexo conjugado  $z_{zi}^*$ , já que para obtenção dos coeficientes reais  $b_i$  do filtro no final da otimização, deve-se expandir o produtório da função de transferência na forma pólo-zero-ganho em (2.36) para obtenção de uma função de transferência direta na forma (2.8) com os  $b_i$  coeficientes reais do filtro.

## 5.6 OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

O termo da língua inglesa *swarm* pode ser descrito como uma população de elementos interativos que são capazes de otimizar um objetivo global através da busca colaborativa do espaço de soluções. Interações locais entre a população de elementos são geralmente enfatizadas. Existe em geral uma tendência estocástica na população dos indivíduos de se moverem para o centro de massa nas dimensões críticas dessa população, resultando na convergência para um ótimo [50].

A partir da idéia de uma população de elementos interativos, em [51] foi proposto um algoritmo paralelo de busca estocástico. Nesse algoritmo, a otimização por enxame de partículas (OEA), do inglês *particle swarm optimization*, uma população de indivíduos são inicializados com estimativas de soluções da função objetivo. Os indivíduos nessa população são chamados de partículas. Cada uma dessas partículas busca uma posição ótima, modelando, por exemplo [52], o comportamento de aves em busca de alimento que são guiadas através do espaço do problema pelos indivíduos ótimos. A posição de cada partícula é atualizada por uma nova velocidade, que por sua vez é baseada na velocidade anterior da partícula, a posição na qual a melhor solução até o presente momento foi alcançada pela partícula e a posição na qual a melhor solução até o presente momento foi alcançada por toda população.

Uma característica marcante na OEA se deve ao fato de que, diferentemente de outras técnicas de otimização evolucionárias, poucos parâmetros necessitam ser ajustados no algoritmo [53]. Entre eles encontram-se o tamanho da população, o momento de inércia  $\alpha$  das partículas e a constante  $\varphi = \varphi_1 + \varphi_2$ , conforme explanados no item 5.6.1. Além disso, para performance com parâmetros ótimos da OEA, a metodologia exploratória estatística do Capítulo 6 será empregada para escolher o valor ótimo dos parâmetros  $\varphi$  e  $\alpha$ .

O restante dessa seção apresenta as equações que determinam a posição e a velocidade da partícula no espaço do problema, subseção 5.6.1, valores que são atualizados à cada iteração do algoritmo OEA com o objetivo de conduzir as partículas para o mínimo da função objetivo de interesse. Adicionalmente, a subseção 5.6.2 apresenta a topologia da vizinhança das partículas, já que a atualização da posição de cada partícula no algoritmo OEA é diretamente relacionada à posição da partícula de maior função de adequação em sua vizinhança.

Por fim, convém notar que a implementação algorítmica da otimização por enxame de partículas, uma população de elementos interativos que são capazes de otimizar um objetivo global através da busca colaborativa do espaço de soluções, com a finalidade de minimização da função objetivo (2.47) do projeto de filtros FIR de fase não-linear, se encontra na subseção I.2.3 do apêndice I.

### 5.6.1 Fórmulas de atualização da velocidade e posição das partículas

O algoritmo OEA mantém uma população de  $M$  partículas durante a otimização. Cada  $i$ -ésima partícula é representada pelo vetor  $\vec{X}_i = (X_{i1}, X_{i2}, \dots, X_{iD})$ , onde  $D$  é a dimensão do

espaço de soluções e  $i = 1, 2, \dots, M$ . Cada  $i$ -ésima partícula possui memória da sua melhor posição (maior função de adequação) anterior  $\vec{P}_i = (P_{i1}, P_{i2}, \dots, P_{iD})$  e de sua velocidade vetorial em cada dimensão  $\vec{V}_i = (V_{i1}, V_{i2}, \dots, V_{iD})$ . Além dessas quantias, cada partícula faz parte de uma vizinhança (ver subseção 5.6.2), e à essa vizinhança é associada a posição da partícula com melhor função de adequação  $\vec{P}_g$ . A cada iteração do algoritmo OEA, os vetores  $\vec{P}_g$  e  $\vec{P}$  de cada partícula da população são combinados para ajustar a velocidade  $\vec{V}$  da partícula e essa mesma velocidade é utilizada para calcular uma nova posição  $\vec{P}$  [54].

Dessa forma a partir das definições acima, a melhor posição da  $i$ -ésima partícula é definida na equação (5.15)

$$P_i^d(t+1) = \begin{cases} P_i^d(t) & \text{se } f(\vec{X}_i(t+1)) \geq f(\vec{P}_i(t)), \\ X_i^d(t+1) & \text{se } f(\vec{X}_i(t+1)) < f(\vec{P}_i(t)) \end{cases}, \quad (5.15)$$

onde  $t$  denota o índice da iteração do algoritmo OEA,  $d$  o índice da dimensão do espaço de soluções do problema e  $f$  a função objetivo de interesse na otimização.

A atualização da velocidade e posição da  $i$ -ésima partícula são definidas para o algoritmo OEA em números contínuos, respectivamente nas equações (5.16) e (5.17) [50].

$$V_i^d(t+1) = \alpha V_i^d(t) + \varphi_1 (P_i^d(t) - X_i^d(t)) + \varphi_2 (P_g^d(t) - X_i^d(t)) \quad (5.16)$$

$$X_i^d(t+1) = X_i^d(t) + V_i^d(t+1), \quad (5.17)$$

onde  $\alpha$  é o momento de inércia da velocidade da partícula,  $\varphi_1$  e  $\varphi_2$  são os parâmetros de controle, também chamados de constante de aceleração.  $\alpha$  é definido em  $0 \leq \alpha < 1$  e  $\varphi_1$  e  $\varphi_2$  são números aleatórios com distribuição uniforme  $\varphi_1, \varphi_2 \sim U(0, 2)$  [50].

Especificamente nesse trabalho, cada  $i$ -ésima posição  $\vec{X}_i$ , melhor posição no espaço de solução  $\vec{P}_i$  e velocidade vetorial  $\vec{V}_i$  da partícula são convertidos em respectivos  $i$ -ésimos zeros  $z_{zi}$  da função de transferência (2.36). Novamente, convém lembrar que tais zeros são definidos no domínio complexo  $z_{zi} \in \mathbb{C}$  e aparecem na função transferência (2.36) em pares complexos conjugados. Dessa forma, na implementação da OEA para resolução do problema de aproximação no projeto de filtros FIR de fase não-linear, cada um dos vetores mantidos pela partícula  $\vec{X}_i$ ,  $\vec{P}_i$  e  $\vec{V}_i$  são representados por dois números de ponto flutuante, respectivamente o módulo  $|z_{zi}|$  e fase  $\angle z_{zi}$  dos zeros da função de transferência em coordenadas polares (2.38). Além disso, para cada  $i$ -ésimo zero  $z_{zi}$  da função de transferência é mantido internamente no algoritmo seu  $i$ -ésimo complexo conjugado  $z_{zi}^*$ , já que para obtenção dos coeficientes reais  $b_i$  do filtro no final da otimização, deve-se expandir o produtório da função de transferência na forma pólo-zero-ganho em (2.36) para obtenção de uma função de transferência direta na forma (2.8) com os  $b_i$  coeficientes reais do filtro.



### 5.6.2 Topologia da vizinhança das partículas

A eficácia do algoritmo OEA se deve à interação entre as partículas com seus vizinhos. Assim que uma partícula descobre um ótimo local, ela se torna a melhor entre seus vizinhos e os mesmos são atraídos para a região ótima. Assim que as partículas se movem para o novo ótimo, elas podem acabar atraindo a primeira partícula em direção à suas melhores posições e assim por diante [50].

Na equação (5.16) de atualização da velocidade da partícula, o termo de influência social das partículas é representado por  $\vec{P}_g$ . Basicamente, essa influência pode ser modelada por duas estruturas genéricas de rede. A primeira é conhecida como topologia do melhor global, já que nessa topologia, cada partícula é atraída para a melhor solução obtida em toda a população [55]. A segunda classificação consiste na topologia local. Nessa topologia a diferença é observada em quais partículas são consideradas vizinhas da  $i$ -ésima partícula e como consequência, exercem um papel primordial na componente social  $\vec{P}_g$  da equação (5.16).

Umas das topologias locais mais conhecida é a topologia em anel. Nela são inclusos apenas os  $k$ -ésimos vizinhos mais próximos da  $i$ -ésima partícula, em termos de seu índice. Geralmente na literatura escolhe-se  $k = 2$ . Uma característica da estrutura em anel com  $k = 2$  é a de oferecer convergência lenta da troca de informações entre as partículas, já que quaisquer duas partículas escolhidas aleatoriamente comunicam informação social através de um caminho longo e indireto. Isso permite as partículas explorarem amplas áreas do espaço de busca sem serem imediatamente guiadas para a melhor partícula global da população [55].

Embora a estrutura em anel possua boa performance em funções com muitos mínimos locais, o problema da escolha da melhor topologia de rede pode ser postulado como um conhecido problema de otimização, o de alocação ótima de tentativas [56]. Além do mais, a performance de várias outras topologias de redes locais foram experimentadas em [56], particularmente com uma técnica de inicialização aleatória de grafos e observou-se que a topologia ótima da vizinhança é dependente do problema (função objetivo). Dessa forma, visando a facilidade de implementação, assim como a tratabilidade na análise do algoritmo OEA na metodologia exploratória estatística do Capítulo 6, optou-se nesse trabalho pela implementação do algoritmo OEA com topologia em anel.





# 6 ANÁLISE EXPLORATÓRIA ESTATÍSTICA DOS ALGORITMOS SOFT-COMPUTING

## 6.1 INTRODUÇÃO

O presente capítulo tem como objetivo apresentar uma metodologia estatística suficientemente abrangente, para aplicação na estimação dos parâmetros ótimos dos algoritmos de otimização *soft-computing* apresentados no Capítulo 5, assim como na criação de experimentos que possibilitem a comparação justa de performance entre os diferentes algoritmos de otimização propostos nesse trabalho para a solução do problema de otimização multi-objetivo no projeto de filtros digitais FIR de fase não-linear.

Conforme anteriormente mencionado, uma problemática no emprego dos algoritmos *soft-computing* consiste na escolha de parâmetros internos desses algoritmos. Exemplos incluem o tamanho da população de indivíduos, taxas de mutação e *crossover*, constantes de aceleração, tamanhos de passo, entre outros. Valores para esses parâmetros são geralmente ajustados através de um processo de tentativa e erro ou baseados em recomendações de problemas relacionados na literatura, embora idealmente, deseje-se uma escolha apoiada em ferramentas estatísticas que ofereçam informações do efeito da variação desses parâmetros na performance dos algoritmos. A metodologia estatística desenvolvida em [35] aborda o problema de como obter a estimativa de parâmetros ótimos. Embora em [35] inicialmente se desenvolva a metodologia para o uso em algoritmos genéticos, conforme mencionado no trabalho, a mesma metodologia é facilmente aplicável em diferentes algoritmos de otimização *soft-computing*.

Seguindo os conceitos estatísticos da metodologia exploratória desenvolvida em [35], experimentos estatísticos adicionais foram desenvolvidos em [29] para abordar o problema de como efetivamente comparar a performance de diferentes algoritmos de otimização *soft-computing* empregados no problema de otimização no projeto de filtros digitais. Com a criação desses experimentos adicionais, a afirmativa de qual é o melhor algoritmo de otimização para determinado problema de otimização não-linear é reforçada por fortes indícios estatísticos, os quais fundamentam o projetista de filtros na escolha do melhor algoritmo de otimização para aplicação em problemas práticos.

Uma possível desvantagem no uso das metodologias estatísticas apresentadas acima consiste na sua aplicação direta em problemas de otimização multi-objetivo, já que indicadores de qualidade mono-objetivo como o valor médio, o desvio-padrão, e momentos de ordem superior da função objetivo são utilizados como a medida de performance comparativa entre as diferentes realizações dos algoritmos *soft-computing*. Dessa forma, pode-se dizer que na otimização mono-objetivo define-se a qualidade por meio da função objetivo, ou seja, quanto menor seu valor, me-

lhora é a solução final obtida. Entretanto, se comparamos duas soluções na presença de múltiplos critérios na otimização, o conceito de dominância Pareto pode ser utilizado, embora deva-se considerar a possibilidade de duas soluções serem incomparáveis, ou seja, ambas não se dominam, complicando consideravelmente a situação [57].

Tendo em vista a problemática de comparação de conjuntos Pareto em problemas multi-objetivos, o presente trabalho propõe o uso de indicadores de qualidade binários, mais especificamente através do indicador  $\epsilon$ -binário [57], o qual é capaz de efetivamente quantificar o quanto um conjunto Pareto  $A$  é melhor que um conjunto  $B$  ou ainda dizer se os dois conjuntos são incomparáveis. Através do indicador  $\epsilon$ -binário, o presente trabalho propõe extensões multi-objetivo para a análise exploratória estatística desenvolvida em [35], com a finalidade de obter parâmetros ótimos para os algoritmos de otimização *soft-computing* em conjunto com o método ponderado de escalarização, aplicados ao problema multi-objetivo de projeto de filtros FIR de fase não-linear. Adicionalmente, serão desenvolvidos experimentos estatísticos que efetivamente comparam a performance de diferentes algoritmos de otimização aplicados ao problema multi-objetivo.

O restante desse capítulo é organizado da seguinte forma. A seção 6.2 apresenta discussão detalhada da análise exploratória estatística mono-objetivo. Já a seção 6.3 apresenta alguns indicadores de qualidade na literatura aplicados a problemas multi-objetivo. Nessa seção também será derivado o indicador  $\epsilon$ -binário, o qual é considerado o melhor indicador no presente momento para a efetiva comparação de performance entre dois conjuntos Pareto [57]. A seção 6.4 apresenta extensões multi-objetivo para análise exploratória estatística da seção 6.2 com o intuito de criar experimentos para estimação de parâmetros ótimos dos algoritmos de otimização *soft-computing* aplicados ao problema de otimização multi-objetivo de projeto de filtros FIR de fase não-linear, assim como a efetiva comparação da performance dos diferentes algoritmos aplicados ao problema de otimização multi-objetivo proposto.

## **6.2 ANÁLISE EXPLORATÓRIA ESTATÍSTICA EM PROBLEMAS DE OTIMIZAÇÃO MONO-OBJETIVO**

Até o presente momento, não existe na literatura uma metodologia amplamente aceita para analisar a relação entre parâmetros e performance para a maioria dos algoritmos *soft-computing*. Levando em consideração os algoritmos apresentados no Capítulo 5, tentar descrevê-los matematicamente é uma tarefa complexa e que apesar dos esforços, ainda não resultou em orientações práticas. Essa situação ocasionou o surgimento de vários estudos que tentam prover tais informações. Entretanto, ambas metodologias e resultados apresentadas na literatura variam muito de uma abordagem para outra [35]. Por essa razão, métodos estatísticos de análise vêm sendo usados para abordar esse problema. Uma das principais desvantagens dos primeiros métodos estatísticos propostos consiste no fato que resultados gerais encontrados nesses métodos variavam muito de uma abordagem para outra. As principais razões para essas notáveis variações nos resultados de

diferentes métodos consistem na falta dos seguintes recursos desejáveis [35]:

1. *Agrupamento em bloco das sementes dos geradores aleatórios como fonte de variação:* Achar conjuntos de parâmetros satisfatórios pode ser difícil devido ao fato que o mesmo conjunto de parâmetros nos mesmos problemas podem levar a diferentes resultados. Na prática, essas diferenças podem ser atribuídas à diferentes sementes na implementação dos geradores aleatórios utilizados na inicialização da população de indivíduos e na implementação dos operadores de modificação aleatória sobre a população do algoritmo. Agrupar as sementes através de unidades experimentais com blocos homogêneos, de tal forma que cada realização do algoritmo para diferentes níveis de seus parâmetros ocorra com as mesmas sementes, limita a causa de variação dentro desses blocos para os parâmetros em estudo. Dessa maneira, variações ou ruídos podem ser reduzidos e as comparações ficam mais precisas [35].
2. *Cálculo do poder e tamanhos amostrais:* Sem o cálculo do poder do número de amostras do experimento estatístico, é incerto se o método estatístico tem poder adequado e assim, tamanho amostral suficiente para detectar diferenças que poderiam ser consideradas importantes. Pequenos tamanhos amostrais geralmente falharão em gerar testes com resultados estatisticamente significativos. Essa situação é particularmente importante se o agrupamento por blocos não é feito já que o conjunto de dados obtidos no experimento fica sujeito a um projeto completamente aleatorizado. Em tal projeto, efeitos podem não ser detectados devido a extensão de ruídos de fundo no conjunto de dados produzidos pelas sementes dos geradores aleatórios. Por essa razão, necessita-se nessas situações de um conjunto de dados extremamente grande para detectar diferenças que sejam de interesse [35].
3. *Análise detalhada da curva de resposta:* É importante realizar uma análise detalhada da curva de resposta do experimento já que ela permite estudar o comportamento do parâmetro através de uma ampla gama de valores implementada.

A fim de decidir sobre o tipo mais adequado de delineamento experimental e teste estatístico, a análise exploratória estatística desenvolvida em [35] aborda os seguintes itens:

1. Agrupamento em bloco para variações devido ao ruído das sementes dos geradores aleatórios.
2. Escolha de um teste estatístico apropriado.
3. Testagem estatística de parâmetros individuais e suas interações.
4. Análise da curva de resposta, permitindo que uma estimativa seja feita dos melhores valores para os parâmetros individuais com intervalos de confiança.
5. Cálculo do poder do experimento estatístico.

Tabela 6.1: Experimento completamente aleatorizado

Bloco	Parâmetro $A$	Parâmetro $B$	Observações
Sementes para réplica do bloco 1	$i$ níveis	$j$ níveis	$ij$
Sementes para réplica do bloco 2	$i$ níveis	$j$ níveis	$ij$
Sementes para réplica do bloco 3	$i$ níveis	$j$ níveis	$ij$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
Sementes para réplica do bloco $n$	$i$ níveis	$j$ níveis	$ij$

Com relação ao primeiro item, o agrupamento em bloco, a metodologia controla o efeito da semente dos gerados aleatórios através da implementação do projeto em blocos completamente aleatorizados. Em tal projeto, cada combinação de níveis dos parâmetros aparece o mesmo número de vezes no mesmo bloco e os blocos são definidos através das sementes. Por exemplo, se existem  $i$  níveis do parâmetro  $A$  e  $j$  níveis do parâmetro  $B$ , então cada bloco contém todas as  $ij$  combinações [35]. Um aumento no tamanho amostral ocorre pela geração de réplicas idênticas dos blocos, exceto pelas sementes. A tabela 6.1 oferece um exemplo de um experimento completamente aleatorizado de  $n$  réplicas e com parâmetros  $A$  e  $B$ .

Com relação ao segundo item da metodologia, a escolha de um teste estatístico apropriado, utiliza-se a Análise da Variância (ANOVA) como meio de comparação de performance para dois ou mais parâmetros usando um projeto em blocos completamente aleatorizados. Na ANOVA, a hipótese nula significa que as médias para diferentes níveis de um parâmetro são iguais. A hipótese alternativa é que as médias para os níveis de um parâmetro não são iguais e que, conclui-se que o parâmetro tem um efeito sobre a variável resposta [35]. Uma característica interessante da ANOVA é que a mesma particiona a variação total das observações contribuídas pelos parâmetros, suas interações, agrupamento em bloco e erro, ou seja, as observações são modeladas pelo seguinte modelo estatístico linear [58]:

$$y_{ij} = \mu + \tau_i + \epsilon_{ij} \begin{cases} i = 1, 2, \dots, a \\ j = 1, 2, \dots, n \end{cases}, \quad (6.1)$$

onde no modelo linear em (6.1)  $\mu$  é um parâmetro comum a todos os níveis do parâmetro do experimento chamado de média global,  $\tau_i$  é um parâmetro único ao  $i$ -ésimo nível do parâmetro,  $\epsilon_{ij}$  é um componente aleatório de erro que incorpora todas as outras fontes de variabilidade no experimento,  $a$  é o número de níveis do parâmetro e  $n$  o número de observações do  $i$ -ésimo parâmetro do experimento. Adicionalmente, o modelo estatístico linear em (6.1) é chamado de modelo da análise da variância de fator único, já que nesse modelo de exemplo apenas um parâmetro é investigado na análise.

Com relação ao erro no modelo estatístico, o mesmo é conceitualizado em termos dos resíduos, os quais são simplesmente os desvios individuais das observações a partir dos valores esperados na suposição que não existe efeito [35], ou seja, define-se o resíduo  $e_{ij}$  da  $j$ -ésima

observação para um  $i$ -ésimo nível do parâmetro do experimento pela seguinte relação [58]:

$$e_{ij} = y_{ij} - \hat{y}_{ij}, \quad (6.2)$$

onde na equação (6.2),  $\hat{y}_{ij}$  é uma estimativa da correspondente observação  $y_{ij}$ . É oportuno observar ainda que, em [58], desenvolve-se detalhadamente as fórmulas para obtenção da estimativa  $\hat{y}_{ij}$ .

O teste para averiguar se um parâmetro é estatisticamente significativo no resultado do experimento é feito na ANOVA da seguinte forma. Primeiramente, divide-se a variação contribuída pelo parâmetro ajustado pelo número de níveis do parâmetro pela variação contribuída pelo erro ajustado pelo número de níveis dos parâmetros e das observações. Esse resultado é uma razão chamada de valor  $F$ , ou seja, se a hipótese nula de que as médias para diferentes níveis de um parâmetro são iguais, o valor  $F$  é definido pela seguinte razão [58]:

$$F = \frac{SQ_{np}/(a-1)}{SQ_E/(N-a)}, \quad (6.3)$$

onde na equação (6.3),  $SQ_{np}$  é o somatório dos quadrados devido aos níveis do parâmetro,  $SQ_E$  é o somatório dos quadrados devido ao erro e  $N = an$ , ou o número total de observações do experimento.

O valor  $F$  em (6.3) é assim chamado já que o mesmo segue a distribuição  $F$  de probabilidade, definida pela seguinte razão [58]:

$$F_{u,v} = \frac{\chi_u^2/u}{\chi_v^2/v}, \quad (6.4)$$

onde na equação (6.4), a distribuição  $F$  de probabilidade possui  $u$  graus de liberdade no seu numerador,  $v$  graus de liberdade no seu denominador e  $\chi_u^2$  e  $\chi_v^2$  são respectivamente duas variáveis aleatórias independentes que seguem distribuição de probabilidade chi-quadrado com graus de liberdade  $u$  e  $v$ .

Dessa forma, conforme visto na equação (6.3), o valor  $F$  possui  $(a-1)$  e  $(N-1)$  graus de liberdade no numerador e denominador. Convém ainda observar que essa equação determina o teste estatístico para a hipótese de que não existem diferenças nas médias dos níveis do parâmetro do experimento (hipótese nula).

O segundo passo para averiguar se um parâmetro é estatisticamente significativo no resultado do experimento, é determinar a probabilidade que se observaria um valor  $F$  tão grande quanto observado sobre a hipótese nula. Esse é o valor  $p$  associado com o valor  $F$  ou simplesmente a probabilidade  $p = \Pr(F)$  [35]. Esse valor geralmente é obtido consultando tabelas padrões com o valor da função densidade de probabilidade de  $F$  com graus de liberdade  $(a-1)$  e  $(N-a)$  [58]. Se o valor  $p$  é igual ou menor que o nível de significância escolhido ( $p \leq 0.01$ ), considera-se que o parâmetro tem um efeito estatístico significativo na variável resposta.

Quanto ao terceiro item da metodologia (teste estatístico para os parâmetros individuais e suas interações), a ANOVA permite que tanto a significância dos parâmetros individuais quanto a interação dos parâmetros sejam estatisticamente testadas. Interação nesse contexto é simplesmente a falha de um parâmetro para produzir o mesmo efeito na variável resposta em diferentes níveis de outro parâmetro. Examinar a interação é importante já que uma interação significativa significa que o efeito de cada parâmetro não pode ser considerado independentemente dos outros [35].

Já o quarto item, a análise da curva de resposta, assim que um parâmetro é demonstrado como estatisticamente significativo na ANOVA, o efeito desse parâmetro pode ser modelado através de um polinômio apropriado. Testes estatísticos podem então ser conduzidos para avaliar se o formato da curva de resposta é predominantemente linear ou é composto de polinômios de alta ordem pelo particionamento da variação de cada parâmetro em termos contrastantes polinomiais ortogonais. Assim que o formato da curva de resposta é estabelecido, regressão polinomial <sup>1</sup> pode ser conduzida para obter estimativas dos coeficientes dos vários parâmetros na equação da curva de resposta [35].

Finalmente, o último item da metodologia, o cálculo do poder do experimento estatístico pode ser feito na ANOVA pelo uso do índice de efeito  $f$  como detalhadamente descrito em [59]. O poder do experimento é expresso pela probabilidade de não-rejeição da hipótese nula quando a hipótese alternativa (a média para diferentes níveis de um parâmetro não são iguais) é verdadeira. Um poder de 80%, quando existe um desvio moderado da hipótese nula é considerado desejável por convenção [59]. Adicionalmente, o poder do experimento é diretamente relacionado ao seu tamanho amostral (número de réplicas). Um tamanho amostral muito pequeno geralmente irá falhar em produzir resultados estatisticamente significativos, enquanto que um tamanho amostral muito elevado pode ser difícil de analisar, além de desperdiçar recursos computacionais [35]. Por fim, é importante notar que devido à natureza exploratória dos experimentos estatísticos, em [35] recomenda-se que, ao menos que a significância estatística tenha sido demonstrada na ANOVA para a interação dos parâmetros, deve-se aumentar o tamanho amostral do experimento por um fator de 5 vezes o tamanho inicial até que se detecte um poder de 80% através do cálculo do índice de efeito  $f$ .

## 6.3 INDICADORES DE QUALIDADE EM PROBLEMAS MULTI-OBJETIVO

Conforme visto nos Capítulos 4 e 5, o presente trabalho propõe o uso de vários algoritmos para a resolução do problema de otimização multi-objetivo no projeto de filtros FIR de fase não-linear, e nós certamente estamos interessados na técnica que oferece as melhores aproximações

---

<sup>1</sup>Suponha que exista uma única variável dependente ou variável de resposta  $y$ , que depende em  $k$  variáveis regressoras independentes  $x_1, x_2, \dots, x_k$ . A relação entre essas variáveis é caracterizada por um modelo matemático chamado de modelo de regressão. Esse modelo é construído para ajustar-se ao conjunto de dados do experimento sobre análise.

para o problema proposto.

A observação acima, por sua vez, leva diretamente ao problema de como avaliar a performance de otimizadores multi-objetivo. Conforme mencionado anteriormente nesse trabalho, se nós comparamos duas soluções na presença de múltiplos critérios de otimização, o conceito de dominância Pareto pode ser utilizado, embora que a possibilidade de duas soluções serem incomparáveis complique a situação. Entretanto, pode-se complicar ainda mais quando comparamos dois conjuntos de soluções, porque algumas soluções em ambos conjuntos podem não ser dominadas pelas soluções no outro conjunto, enquanto que outras soluções podem ser incomparáveis [57]. Consequentemente, não é claro o conceito de qualidade com respeito à aproximações do conjunto Pareto-ótimo: proximidade com relação às soluções ótimas no espaço objetivo, cobertura de uma ampla gama de soluções diversas, entre outros possíveis critérios [57].

Vários estudos que abordam o problema de comparação de aproximações do conjunto Pareto ótimo de uma maneira quantitativa podem ser encontrados na literatura. Os mais populares são as medições unárias de qualidade, por exemplo, uma medida que atribua para cada conjunto de aproximação um número que reflita um certo aspecto de qualidade, e usualmente uma combinação desses indicadores é utilizada. Outros métodos são baseados em medições binárias de qualidade, as quais atribuem números para pares de conjuntos de aproximação. Em [57], uma metodologia matemática é desenvolvida para avaliar corretamente as diferentes técnicas de medição propostas na literatura e as seguintes afirmações são provadas para dois conjuntos Pareto de aproximação arbitrários  $A$  e  $B$  [57]:

- Não existe medição unária de qualidade que seja capaz de indicar se um conjunto de aproximação  $A$  é melhor que um conjunto de aproximação  $B$ .
- A afirmação acima ainda é válida se consideramos uma combinação finita de medições unárias.
- A maioria das medições de qualidade que foram propostas para indicar que  $A$  é melhor que  $B$  na melhor das hipóteses permitem inferir que  $A$  não é melhor que  $B$ , por exemplo,  $A$  é melhor que ou incomparável a  $B$ .
- Medições binárias de qualidade superam as limitações das medições unárias e, se apropriadamente projetadas, são capazes de indicar se um conjunto  $A$  é melhor que  $B$ .

Baseado nas afirmações acima, o presente trabalho propõe o uso da classe de indicadores binários de qualidade como meio de medição de performance no problema multi-objetivo proposto, já que essa classe de indicadores é capaz de dizer se um conjunto Pareto de aproximação  $A$  é melhor que  $B$ . Antes de propriamente apresentar o indicador binário de qualidade proposto, é oportuno mencionar as relações matemáticas disponíveis para conjuntos de aproximação desenvolvidas em [57], as quais são:



- *Domina estritamente*  $\succ\succ$ : Por exemplo,  $A \succ\succ B$ , cada  $z^2 \in B$  é estritamente dominado por pelo menos um  $z^1 \in A$ .
- *Domina*  $\succ$ : Por exemplo,  $A \succ B$ , cada  $z^2 \in B$  é dominado por pelo menos um  $z^1 \in A$ .
- *Melhor*  $\triangleleft$ : Por exemplo,  $A \triangleleft B$ , cada  $z^2 \in B$  é fracamente dominado por pelo menos um  $z^1 \in A$  e  $A \neq B$ .
- *Domina fracamente*  $\succeq$ : Por exemplo,  $A \succeq B$ , cada  $z^2 \in B$  é fracamente dominado por pelo menos um  $z^1 \in A$ .
- *Incomparável*  $\parallel$ : Por exemplo,  $A \parallel B$ , nenhum  $A$  fracamente domina  $B$  nem  $B$  domina fracamente  $A$ .

, onde nas relações matemáticas acima,  $z^1$  e  $z^2$  são dois vetores de funções objetivo quaisquer.

A partir das relações acima, em [57] o indicador  $\epsilon$ -binário,  $I_\epsilon(A, B)$ , é definido por:

$$I_\epsilon(A, B) = \max_{z^2 \in B} \min_{z^1 \in A} \max_{1 \leq i \leq n} \frac{z_i^1}{z_i^2}, \quad (6.5)$$

onde  $z_i^n$  é a  $i$ -ésima função objetivo de um vetor de funções objetivos arbitrário  $z^n$ .

Com esse indicador, em [57] mostra-se que  $I_\epsilon(B, A) > 1$  implica, na maioria das vezes, em  $A \triangleleft B$ . Entretanto, são apresentadas situações onde  $I_\epsilon(B, A) > 1$  mesmo quando os dois conjuntos são incomparáveis, ou seja, o indicador  $\epsilon$ -binário implica em uma das seguintes propriedades quando  $I_\epsilon(B, A) > 1$ :

1.  $A \triangleleft B$ .
2.  $A \parallel B$ .

Entretanto, usando o seguinte teste  $F : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \{\text{verdadeiro}, \text{falso}\}$ :

$$F = (I_\epsilon(A, B) \leq 1 \wedge I_\epsilon(B, A) > 1), \quad (6.6)$$

prova-se em [57] que somos capazes de concluir que  $A$  é melhor do que  $B$ ,  $A \triangleleft B$ .

Devido às interessantes propriedades citadas acima, o presente trabalho utiliza o indicador  $\epsilon$ -binário como meio de avaliar a performance das aproximações dos conjuntos Pareto encontrados para a soluções do problema multi-objetivo do projeto de filtros FIR de fase não-linear.

Como última observação, faz-se notar que a equação (6.5) não possui aplicação direta, sendo utilizada na prática por meio do algoritmo 6.3.1 [60]:

, onde no algoritmo 6.3.1, o operador  $|\cdot|$  denota o tamanho do conjunto de aproximação.

---

**Algoritmo 6.3.1:** Implementação do indicador  $\epsilon$ -binário

---

- 6.3.1.1 Calcule a matriz  $\epsilon_{i,j} = \max_{1 \leq k \leq n} \left( \frac{z_k^i}{z_k^j} \right) \forall i \in \{1, \dots, |\mathbf{A}|\} \wedge \forall j \in \{1, \dots, |\mathbf{B}|\}$  ;
- 6.3.1.2 Calcule o vetor  $\epsilon_{min} = \min_{1 \leq i \leq |\mathbf{A}|} (\epsilon_{i,j})$ ;
- 6.3.1.3 Calcule o indicador  $\epsilon$ -binário  $I_\epsilon(\mathbf{A}, \mathbf{B}) = \max_{1 \leq i \leq |\mathbf{B}|} (\epsilon_{min})$ ;
- 

## 6.4 EXTENSÕES MULTI-OBJETIVO PARA A ANÁLISE EXPLORATÓRIA ESTATÍSTICA

### 6.4.1 Estimação dos parâmetros ótimos dos algoritmos de otimização

Tendo apresentado a análise exploratória estatística da seção 6.2 e o indicador  $\epsilon$ -binário da seção 6.3, pode-se agora definir uma análise exploratória estatística, com extensão a problemas multi-objetivos, tendo a finalidade de estimação dos parâmetros ótimos para os algoritmos *soft-computing* aplicados ao problema de projeto de filtros FIR de fase não-linear.

Os passos para realizar a estimação ótima de parâmetros são definidos por:

1. Criar um projeto completamente aleatorizado para os algoritmos *soft-computing* em estudo.
2. Gerar um diagrama de pontos para minimizar a ocorrência de observações censuradas.
3. Finalizar os valores iniciais dos parâmetros para a análise exploratória estatística.
4. Gerar um conjunto de dados inicial consistindo de um número arbitrário de réplicas.
5. Calcular o poder baseado em um valor de efeito pré-determinado.
6. Conduzir a análise da variância (ANOVA) e determinar quais parâmetros são estatisticamente significativos.
7. Para parâmetros que forem estatisticamente significativos, particionar o somatório dos quadrados em termos contrastantes polinomiais. Determinar quais termos polinomiais são estatisticamente significativos.
8. Usar regressão polinomial para obter os coeficientes para a curva de resposta.
9. Diferenciar e resolver a curva de resposta de cada parâmetro para obter os melhores valores e calcular intervalos de confiança.

Com relação ao primeiro item da análise, os seguintes experimentos serão criados nesse trabalho para os algoritmos *soft-computing* apresentados no Capítulo 5:

- *Algoritmo genético com método ponderado de escalarização*: Estimação ótima das taxas de mutação e *crossover* com um experimento em blocos completamente aleatorizados definido na tabela 6.2.

Tabela 6.2: Experimento em blocos completamente aleatorizados para o algoritmo genético

Bloco	Taxa de Mutação	Taxa de Crossover	Observações
Sementes para réplica do bloco 1	$i$ níveis	$j$ níveis	$ij$
Sementes para réplica do bloco 2	$i$ níveis	$j$ níveis	$ij$
Sementes para réplica do bloco 3	$i$ níveis	$j$ níveis	$ij$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
Sementes para réplica do bloco $n$	$i$ níveis	$j$ níveis	$ij$

Tabela 6.3: Experimento em blocos completamente aleatorizados para a otimização por enxame de partículas

Bloco	Parâmetro $\phi$	Momento de inércia	Observações
Sementes para réplica do bloco 1	$i$ níveis	$j$ níveis	$ij$
Sementes para réplica do bloco 2	$i$ níveis	$j$ níveis	$ij$
Sementes para réplica do bloco 3	$i$ níveis	$j$ níveis	$ij$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
Sementes para réplica do bloco $n$	$i$ níveis	$j$ níveis	$ij$

- *Otimização por enxame de partículas com método ponderado de escalarização*: Estimação ótima do parâmetro  $\phi$  e do momento de inércia  $\alpha$  com experimento em blocos completamente aleatorizados definidos na tabela 6.3.

Seguindo as diretrizes da análise exploratória estatísticas mono-objetivo em [35], a implementação dos algoritmos acima foi deliberadamente simples. Dessa forma, parâmetros como, por exemplo, o tamanho da população não foram variados mas mantidos com valores recomendados da literatura. Somente dois parâmetros de cada algoritmo foram estatisticamente analisados e otimizados. A principal razão para isso se deve ao fato que com mais de dois parâmetros, a análise estatística se torna mais complexa e computacionalmente cara.

O segundo passo na metodologia consiste na geração do diagrama de pontos para minimizar a ocorrência de censura. Para alcançar esse objetivo, deve-se gerar um experimento com um número arbitrário de réplicas (na análise exploratória mono-objetivo usa-se um valor de 10 réplicas [35]) usando o intervalo permitido de valores para os parâmetros dos algoritmos *soft-computing*, por exemplo, para o experimento em blocos completamente aleatorizados do algoritmo genético da tabela 6.2 utilizar o parâmetro *crossover*  $p_{cr}$  entre  $0 < p_{cr} < 1$  e o parâmetro *mutação*  $p_{mu}$  entre  $0 < p_{mu} < 1$ . Uma instância de censura é identificada nesse experimento quando a diferença entre o indicador de qualidade do melhor indivíduo na população do algoritmo menos o indicador de qualidade do pior indivíduo é menor ou igual a um valor limiar arbitrário. Quando essa situação ocorre, um ponto é inserido no diagrama, conforme pode ser visto no exemplo da Figura 6.1.

Com relação ao indicador de qualidade  $\epsilon$ -binário, dado que o mesmo deve ser comparado

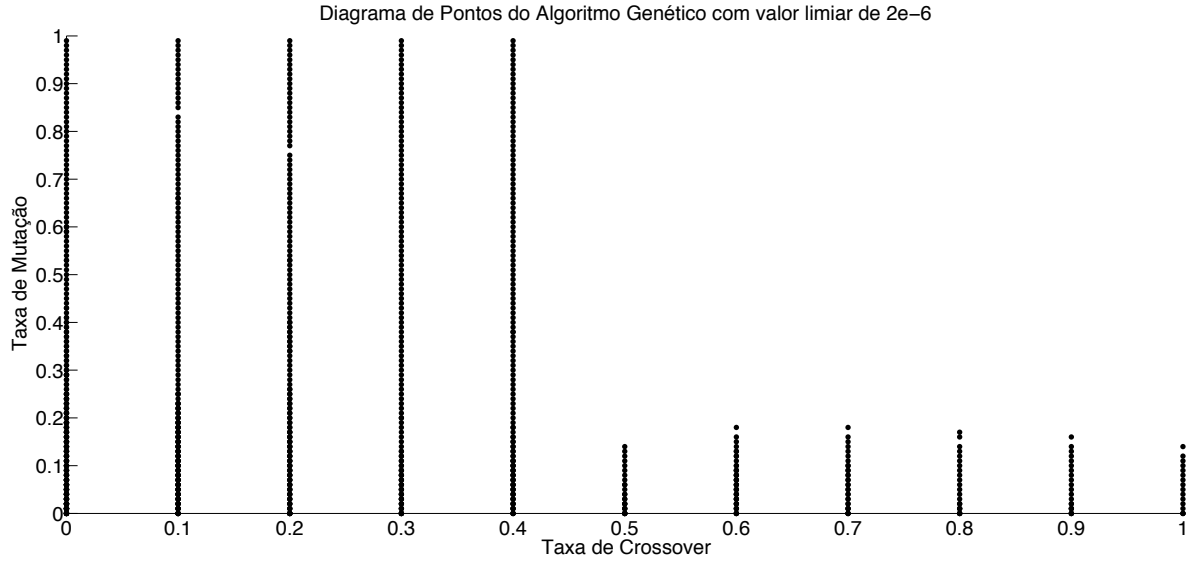


Figura 6.1: Exemplo de diagrama de pontos para o algoritmo genético

a outro conjunto de aproximação, o presente trabalho propõe o uso de um conjunto arbitrário de referência  $\mathbf{Z}^*$ , no qual todos os algoritmos devem ser comparados. Mais especificamente, o conjunto referência é composto por apenas um único ponto, uma estimativa do vetor objetivo ideal  $\mathbf{z}^*$ . Defini-se  $\mathbf{z}^*$  pelo vetor que minimiza individualmente cada uma das funções objetivos. Dessa forma, os componentes desse vetor são obtidos pela minimização individual de cada uma das funções objetivos do problema proposto. Convém notar ainda que, para o problema de otimização multi-objetivo do projeto de filtros FIR de fase não-linear, as funções objetivos  $f_1$ , equação (3.31) e  $f_2$ , equação (3.32) do problema, respectivamente a distância de aproximação da atenuação em decibéis e a distância de aproximação do atraso de grupo em amostras do filtro FIR projetado, são funções não-convexas, e como a definição do vetor objetivo ideal assume que possuímos conhecimento do mínimo global da funções objetivos individuais [19],  $\mathbf{z}^*$  é na verdade apenas uma estimativa do vetor objetivo ideal já que não possuímos a garantia de obter o mínimo global na solução das funções objetivo. Escolheu-se essa abordagem de comparação do conjunto de referência  $\mathbf{Z}^*$  com os conjuntos de aproximação obtidos devido ao fato que na maioria das vezes essa comparação é válida para o indicador  $\epsilon$ -binário  $I_\epsilon(\mathbf{Z}^*, \mathbf{A})$ , ou seja,  $\mathbf{Z}^* \triangleleft \mathbf{A}$ .

Considerando as observações acima, o teste realizado para obter  $I_\epsilon(\mathbf{Z}^*, \mathbf{A})$  é definido no algoritmo 6.4.1.

---

**Algoritmo 6.4.1:** Teste para obtenção do indicador  $\epsilon$ -binário

---

6.4.1.1 se  $(I_\epsilon(\mathbf{Z}^*, \mathbf{A}) \leq 1 \wedge I_\epsilon(\mathbf{A}, \mathbf{Z}^*) > 1) = TRUE$  então

6.4.1.2     |  $valor_{indicador} = I_\epsilon(\mathbf{Z}^*, \mathbf{A})$  ;

6.4.1.3 senão

6.4.1.4     | descartar conjunto de aproximação  $\mathbf{A}$  da análise estatística;

---

Dando prosseguimento à metodologia, o próximo passo consiste na finalização dos intervalos iniciais dos parâmetros da análise exploratória estatística. Para realizar essa tarefa, os melhores valores dos parâmetros são encontrados para um número arbitrário de experimentos realizados com os valores dos parâmetros não censurados do diagrama de pontos (na análise exploratória estatística mono-objetivo utiliza-se 10 experimentos com 100 réplicas cada). A combinação de parâmetros resultando na melhor performance é achada em cada experimento. Quando essas combinações são agrupadas, por exemplo 10 combinações na análise exploratória estatística mono-objetivo [35], elas demonstram os níveis inferiores e superiores dos parâmetros associados com a melhor performance.

Após achar os valores finais de parâmetros do passo acima, um experimento em blocos completamente aleatorizado de 100 réplicas [35] deve ser gerado para cada algoritmo. Para esse experimento fatorial, ao invés de calcular separadamente o somatório dos quadrados para os efeitos principais e para o somatório total dos quadrados, geralmente utiliza-se de *software* especializado, por exemplo S-PLUS ®, para conduzir a ANOVA em um experimento fatorial com dois fatores [58]. O resultado desse procedimento é geralmente uma tabela da ANOVA, a qual mostra o somatório particionado dos quadrados para cada fator do experimento, por exemplo, para o algoritmo genético, o *crossover*, mutação e a interação de ambos. Conforme explanado em [35], os valores dos somatórios particionados dos quadrados são necessários para avaliar o grau dos polinômios usados na regressão linear para obtenção da curva de resposta. A partir dessa curva de resposta, valores ótimos dos parâmetros podem ser estimados. Além disso, a tabela ANOVA apresenta o nível de significância que é utilizado para inspecionar se o parâmetro de interação é estatisticamente significativo, por exemplo, na análise exploratória estatística mono-objetivo em [35],  $p \leq 0.01$ . Caso não haja significância estatística no parâmetro de interação, não é necessário aumentar o tamanho amostral (número de réplicas) do experimento.

Após obter o somatório particionado dos quadrados no passo acima da metodologia, regressão linear deve ser usada para estimar a curva/superfície de resposta para cada algoritmo em estudo. Um exemplo de superfície de resposta é ilustrada na Figura 6.2.

Finalmente, o último passo da metodologia consiste em obter a estimativa dos parâmetros ótimos, minimizando a superfície de resposta encontrada no passo acima.

#### **6.4.2 Comparação de performance entre os algoritmos de otimização**

Adicionalmente a obtenção de parâmetros ótimos para os algoritmos *soft-computing* conforme a metodologia desenvolvida na subseção 6.4.1, certamente estamos interessados na obtenção da técnica que oferece a melhor aproximação do conjunto Pareto-ótimo para o problema de otimização multi-objetivo do projeto de filtros FIR de fase não-linear. Com essa finalidade, um experimento em blocos completamente aleatorizados é adicionalmente criado de acordo com a tabela 6.4.

Para efetivamente comparar os algoritmos de otimização por meio do indicador de qualidade

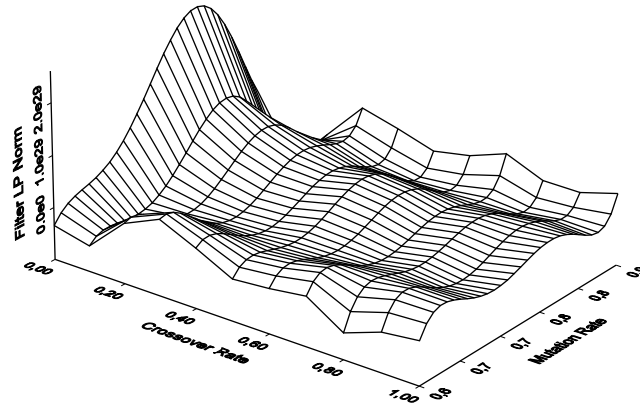


Figura 6.2: Exemplo de superfície de resposta para algoritmo genético [29]

Tabela 6.4: Experimento em blocos completamente aleatorizados para comparação dos algoritmos de otimização

Bloco	Algoritmo de Otimização	Observações
Sementes para réplica do bloco 1	$i$ níveis	$i$
Sementes para réplica do bloco 2	$i$ níveis	$i$
Sementes para réplica do bloco 3	$i$ níveis	$i$
$\vdots$	$\vdots$	$\vdots$
Sementes para réplica do bloco $n$	$i$ níveis	$i$

$\epsilon$ -binário definido a partir do teste do algoritmo 6.3.1, os seguintes passos devem ser executados:

1. Criar um projeto em blocos completamente aleatorizados com os algoritmos de otimização em estudo, conforme o experimento da tabela 6.4.
2. Atribuir os parâmetros ótimos para os algoritmos de otimização conforme a metodologia da subseção 6.4.1.
3. Calcular as estatísticas dos indicadores binários de qualidade para cada algoritmo, por exemplo, momentos de primeira e segunda ordem para  $I_\epsilon(\mathbf{Z}^*, \mathbf{A})$ .
4. O melhor algoritmo para o problema proposto é o que possuir o maior valor dos momentos estatísticos de  $I_\epsilon(\mathbf{Z}^*, \mathbf{A})$ . Por exemplo, o maior valor da média  $\overline{I_\epsilon(\mathbf{Z}^*, \mathbf{A})}$ .

Por fim, é oportuno discutir uma última questão com relação ao experimento estatístico de comparação de performance dos algoritmos de otimização. Claramente, o máximo que se pode

afirmar sobre o experimento estatístico é que o algoritmo obtido possui melhor performance apenas para o conjunto de especificações do filtro FIR utilizado na geração do experimento. Por exemplo, se para um experimento estatístico hipotético do projeto de um filtro FIR passa-faixa com determinado conjunto de especificações, o melhor algoritmo obtido foi o algoritmo genético, não podemos a priori generalizar essa afirmação para o projeto de um filtro rejeita-faixa. Isso se deve ao fato que a função objetivo para otimização muda de um conjunto de especificações para outro no projeto do filtro.

Com relação à questão de qual o melhor algoritmo de otimização que pode ser encontrado para um determinado domínio de aplicação, uma série de trabalhos interessantes foram publicados. Mais especificamente, o teorema *no free lunch* - NFL, diz que para qualquer par de algoritmos  $a_1$  e  $a_2$ , a seguinte relação é válida [61]:

$$\sum_f P(d_m^y | f, m, a_1) = \sum_f P(d_m^y | f, m, a_2), \quad (6.7)$$

onde na equação (6.7),  $d_m^y$  denotam todas as amostras do valor da função objetivo  $f$  e o operador  $P$  na equação denota a probabilidade condicional de se obter uma amostra particular  $d_m$  sobre as condições impostas.

A equação em (6.7) diz que a performance média de quaisquer pares de algoritmos sobre todos os possíveis problemas de otimização é idêntica. Isso significa que se um algoritmo  $a_1$  tem performance superior do que outro algoritmo  $a_2$  em algum conjunto de problemas, então o inverso tem que ser verdadeiro para todo o conjunto de problemas de otimização restantes [61].

É desejável que um algoritmo tenha boa performance em todas as funções possíveis, mas felizmente, na prática, isso é conveniente somente para um subconjunto que resulta em problemas práticos. Recentemente, uma versão restrita do teorema NFL mostrou que os resultados do teorema NFL em (6.7) se mantêm para um subconjunto  $F$  de todas as funções possíveis se e somente se  $F$  é fechado sobre permutação e cada função alvo em  $F$  é igualmente provável [62]. Adicionalmente, em [62] mostra-se que a fração de subconjuntos não-vazios que são fechados sobre permutação decai rapidamente a zero assim que a cardinalidade do espaço de busca aumenta, e que restrições na variável de controle e número de mínimos locais levam a subconjuntos que não são fechados sobre permutação [63].

Esses resultados são uma motivação para investigar o fato de que para problemas práticos, por exemplo o problema de otimização no projeto de filtros digitais, espera-se que alguns algoritmos possuam melhor performance, e que alguma generalização com relação ao melhor algoritmo possa ser feita nesse subconjunto de possíveis funções. Essa questão é deixada como trabalho futuro e não será investigada no presente trabalho.



# 7 RESULTADOS

## 7.1 INTRODUÇÃO

O presente capítulo tem por objetivo apresentar alguns resultados obtidos com a formulação multi-objetivo do projeto de filtros digitais FIR de fase não-linear. Conforme discutido anteriormente, o presente trabalho propõe o emprego de ambos algoritmos de otimização baseados no gradiente (Quasi-Newton), assim como de algoritmos *soft-computing* (Algoritmo Genético, Resfriamento Simulado e Otimização por Enxame de Partículas) aplicados na minimização simultânea das funções objetivos  $f_1$ , equação (3.31) e  $f_2$ , equação (3.32), respectivamente as distâncias de aproximação entre a magnitude da resposta em frequência e atraso de grupo do filtro projetado e das especificações desejadas impostas pelo projetista.

Primeiramente, seguindo a metodologia exploratória estatística do Capítulo 6, experimentos estatísticos serão computacionalmente implementados para obtenção de estimativas de parâmetros ótimos para os algoritmos *soft-computing* aplicados em um projeto de filtro FIR rejeita-faixa arbitrário, conforme os passos detalhados na subseção 6.4.1. Logo em seguida, com as estimativas de parâmetros ótimos, todos os algoritmos de otimização propostos no presente trabalho serão comparados a nível de performance aplicados novamente em um projeto de filtro FIR rejeita-faixa arbitrário, conforme os passos detalhados na subseção 6.4.2. A finalidade desses experimentos estatísticos é obter o melhor algoritmo de otimização para o projeto de teste arbitrário proposto.

Adicionalmente, os algoritmos de otimização serão sistematicamente empregados no projeto de algumas especificações de filtros ideais: passa-faixa e rejeita-faixa, todos com atraso de grupo reduzido na banda de passagem/rejeição de interesse. Análises do conjunto Pareto-ótimo para os filtros resultantes serão feitas abordando considerações de compromisso entre a aproximação simultânea das especificações de magnitude e de atraso de grupo do filtro.

O restante desse capítulo é organizado da seguinte forma. A seção 7.2 apresenta descrição detalhada das ferramentas computacionais de suporte utilizadas na geração dos resultados desse capítulo. Já a seção 7.3 apresenta resultados para a análise exploratória estatística do Capítulo 6 aplicadas no projeto de um filtro FIR rejeita-faixa arbitrário com atraso de grupo reduzido. Mais especificamente, a subseção 7.3.1 apresenta estimativa de parâmetros ótimos para ambos Algoritmo Genético e Otimização por Enxame de Partículas. A subseção 7.3.2 apresenta resultados do comparativo de performance entre todos os algoritmos de otimização propostos nesse trabalho (Quasi-Newton, Algoritmo Genético, Resfriamento Simulado e Otimização por Enxame de Partículas). Já a seção 7.4 apresenta discussão detalhada do emprego dos algoritmos propostos no projeto de filtros usuais na área de processamento digital de sinais: respectivamente, rejeita-faixa e passa-faixa. Considerações de compromisso entre a aproximação simultânea das especificações de magnitude e de atraso de grupo do filtro que o projetista deve considerar serão abordadas.



## 7.2 FERRAMENTAS COMPUTACIONAIS DE SUPORTE

### 7.2.1 Framework de Otimização

A geração dos resultados do presente trabalho necessita da implementação computacional dos algoritmos de otimização propostos, mecanismos de geração de experimentos estatísticos, implementação da função multi-objetivo e especificação dos filtros para o projeto de filtros FIR com a aproximação arbitrária da magnitude e atraso de grupo e de mecanismos simples de configuração dos algoritmos e das especificações dos filtros projetados. Com essa finalidade, foi desenvolvido um *software* na plataforma Java [64] que oferece suporte as funcionalidades citadas acima.

Um princípio de projeto na criação do *software* para geração dos resultados do presente trabalho foi a de permitir a fácil reutilização dos algoritmos de otimização, função objetivo e experimentos estatísticos, de tal forma que o *software* final desenvolvido seja facilmente reutilizado em outros domínios de aplicação. Seguindo uma metodologia de análise e projeto de *software* orientados a objetos, foi desenvolvido um *framework* de otimização que atende esses objetivos.

Uma definição de *framework* é a de um conjunto de classes que tornam possível sua reutilização para um determinado domínio de aplicação. Um *framework* provém guia arquitetônico pelo particionamento do projeto em classes abstratas e pela definição de suas responsabilidades e colaborações. Um desenvolvedor customiza o *framework* para uma aplicação particular através da herança das classes abstratas e pela composição de instâncias das classes do *framework* [65].

Para exemplificar a estrutura das classes que compõe o *framework* de otimização desenvolvido, é oportuno utilizar-se da UML (*Unified Modeling Language*), Linguagem de Modelagem Unificada, uma linguagem de modelagem amplamente utilizada na documentação e especificação de *softwares* orientados a objeto. Essa linguagem propõe-se à modelar a arquitetura de sistemas através de cinco visões: visão de projeto, implementação, processo, distribuição e caso de uso [66]. Tais visões são representadas na linguagem através de diagramas, uma representação gráfica de conjunto de elementos. Os diagramas da UML que definem as cinco visões são [66]:

- Diagrama de Classe.
- Diagrama de Objeto.
- Diagrama de Caso de Uso.
- Diagrama de Seqüência.
- Diagrama de Colaboração.
- Diagrama de Estados.
- Diagrama de Atividade.
- Diagrama de Componentes.

- Diagrama de Distribuição.

Para apresentação do *framework*, será utilizado o diagrama de classes. Esse diagrama mostra um conjunto de classes, interfaces, colaborações e seus relacionamentos, e são os mais utilizados na modelagem de *softwares* orientados a objeto [66]. Esses diagramas endereçam a modelagem da visão estática/estrutural do sistema.

Para descrever de forma simplificada o *framework* de otimização, é oportuno dividi-lo em quatro módulos principais:

1. Algoritmos de otimização.
2. Função objetivo e especificação do filtro digital.
3. Experimentos estatísticos em blocos completamente aleatorizados.
4. Mecanismo de configuração dos módulos acima.

A estrutura de classes que suporta o primeiro módulo do *framework*, algoritmos de otimização, pode ser vista na Figura 7.1.

Conforme visto na Figura 7.1, todos os algoritmos propostos no presente trabalho estão representados no diagrama. Mais especificamente, os algoritmos *soft-computing*: algoritmo genético, resfriamento simulado e otimização por enxame de partículas herdam da classe abstrata *AbstractOtimizadorMonoObjetivoSoftComputing*. Por sua vez o algoritmo Quasi-Newton herda da classe abstrata *AbstractOtimizadorMonoObjetivoComGradiente*. Devido ao problema do projeto de filtros FIR de fase não-linear ser inerentemente multi-objetivo, a classe *AlgoritmoPonderadoDeEscalarizacao* implementa o método ponderado de escalarização, o qual por sua vez, utiliza alguma subclasse do algoritmo de otimização mono-objetivo, representado no diagrama por *AbstractOtimizadorMonoObjetivo*, para minimização de cada subproblema de otimização mono-objetivo pela apropriada variação dos pesos, conforme a equação (3.19) de ponderação.

A estrutura de classes que suporta o segundo módulo do *framework*, função objetivo e especificação do filtro digital, pode ser vista na Figura 7.2

Conforme visto na Figura 7.2, esse módulo implementa a especificação e o projeto de filtros digitais FIR, além da postulação do projeto como um problema de otimização através da função objetivo. Mais especificamente, a especificação do filtro digital FIR é feita através das classes *ParDeEspecificacoesRespostaModulo* e *ParDeEspecificacoesAtrasoDeGrupo*. Essas duas classes possuem informação de qual é a norma  $L_p$ , o valor da função de pesos  $W$  e respectivamente, o valor da magnitude da resposta em frequência em decibéis e o atraso de grupo do filtro para cada banda de frequência especificada no projeto. Já as classes *FuncaoCustoFiltroFIRFaseNaoLinearMagnitude* e *FuncaoCustoFiltroFIRFaseNaoLinearAtrasoDeGrupo* implementam respectivamente as equações (3.31) e (3.32) das distâncias de aproximação das repostas de magnitude e atraso de grupo entre o filtro projetado e as especificações desejadas.

A estrutura de classes que suporta o terceiro módulo do *framework*, experimentos estatísticos em blocos completamente aleatorizados, pode ser vista na Figura 7.3.

Conforme visto na Figura 7.3, esse módulo implementa os experimentos da análise exploratória estatística do Capítulo 6. Mais especificamente, as classes *ExperimentoComAlgoritmoGenetico* e *ExperimentoComParticleSwarmOptimization* implementam os experimentos em blocos completamente aleatorizados da subseção 6.4.1, com a finalidade de estimação dos parâmetros ótimos dos algoritmos de otimização. Já a classe *ExperimentoProjetoFiltroFIR* implementa o experimento em blocos completamente aleatorizados da subseção 6.4.2, com a finalidade de comparação justa de performance entre os algoritmos de otimização, aplicados no projeto de um filtro FIR arbitrário.

Por fim, a estrutura que suporta o quarto módulo do *framework*, mecanismo de configuração, pode ser visto na Figura 7.4.

Conforme visto na Figura 7.4, esse módulo tem por objetivo permitir ao usuário do *framework* configurar de forma simplificada os vários parâmetros necessários para geração dos resultados do presente capítulo. Mais especificamente, as configurações feitas pelo usuário são especificadas através de um arquivo de configuração no formato XML (*eXtensible Markup Language*) [67]. Esse formato foi originalmente criado para transmissão de conteúdo na Internet com o objetivo de especificar a estrutura dos documentos transmitidos. Entretanto, o XML pode ser usado com outros propósitos além de descrição de páginas Web, tendo sido utilizado extensivamente como linguagem de comunicação entre *softwares* heterogêneos e na especificação de elementos de configuração em *frameworks* e aplicativos diversos.

Como exemplo do arquivo de configuração do *framework* de otimização desenvolvido, o trecho de arquivo 1 no Anexo II exemplifica a sintaxe para especificação de um filtro FIR rejeita-faixa com atraso de grupo reduzido nas bandas de passagem.

Os trechos do arquivo de configuração 2, 3 e 4 no Anexo II apresentam respectivamente, a sintaxe para configuração do algoritmo genético, resfriamento simulado e otimização por enxame de partículas em conjunção com o método ponderado de escalarização multi-objetivo.

Por fim, o trecho do arquivo de configuração 5 no Anexo II apresenta a sintaxe de configuração do módulo de geração dos experimentos estatísticos em bloco completamente aleatorizados.

## 7.2.2 Análise Exploratória Estatística

Os resultados da análise exploratória estatística são gerados com o suporte de diversas ferramentas computacionais. Primeiramente, os resultados dos respectivos experimentos em blocos completamente aleatorizados para geração do diagrama de pontos, finalização dos intervalos iniciais, otimização dos parâmetros e comparativo de performance dos algoritmos, são gerados pelo *framework* de otimização da subseção 7.2.1. Os resultados desses experimentos são gravados em arquivos texto.

Para geração do diagrama de pontos e finalização dos intervalos iniciais, foram desenvolvidos *scripts* no ambiente MATLAB [68] que fazem a leitura dos arquivos com o resultado dos experimentos, e posteriormente geram o diagrama de pontos e finalizam os intervalos dos parâmetros, conforme processo descrito na subseção 6.4.1.

Para geração da tabela ANOVA dos experimentos de estimação dos parâmetros ótimos dos algoritmos, *scripts* foram implementados no ambiente S-PLUS [69]. Adicionalmente, esses *scripts* geram o somatório particionado dos quadrados, informação que é utilizada para obter a ordem dos polinômios na regressão linear e na geração das curvas/superfícies de resposta dos experimentos de estimação dos parâmetros.

Por fim, as estatísticas do indicador  $\epsilon$ -binário para o experimento de comparação da performance dos algoritmos de otimização foram implementadas através de *scripts* MATLAB que fazem a leitura dos arquivos com o resultado dos experimentos.

### 7.2.3 Análise Gráfica dos Filtros Resultantes

A análise gráfica dos filtros resultantes é feita através da geração de gráficos com o conjunto Pareto-ótimo encontrado em determinado algoritmo aplicado no projeto de um filtro FIR arbitrário, além da geração de gráficos com a magnitude e fase da resposta em frequência, atraso de grupo e diagrama de pólos e zeros do filtro em análise. Todas essas funcionalidades são implementadas através de *scripts* MATLAB que fazem a leitura dos arquivos com o vetor de zeros  $A$  da função transferência do filtro otimizado em coordenadas polares, conforme a equação (2.37).

## 7.3 RESULTADOS DA ANÁLISE EXPLORATÓRIA ESTATÍSTICA

Essa seção tem por objetivo apresentar os resultados da análise exploratória estatística do Capítulo 6. Mais especificamente, a seção 7.3.1 apresenta os resultados da estimação dos parâmetros ótimos para o algoritmo genético e para a otimização por enxame de partículas. Já a seção 7.3.2 apresenta os resultados do comparativo de performance entre os algoritmos de otimização.

A análise exploratória estatística será implementada para o seguinte conjunto de especificações de um filtro rejeita-faixa com atraso de grupo reduzido nas bandas de passagem:

$$\begin{aligned}
W_\eta(\omega) &= 1 && \begin{cases} 0 \leq \omega \leq 0.25 \\ 0.3 \leq \omega \leq 0.65 \\ 0.7 \leq \omega \leq \pi \end{cases} \\
W_\eta(\omega) &= 0 && \begin{cases} 0.25 < \omega < 0.3 \\ 0.65 < \omega < 0.7 \end{cases} \\
W_\tau(\omega) &= 1 && \begin{cases} 0 \leq \omega \leq 0.25 \\ 0.7 \leq \omega \leq \pi \end{cases} \\
W_\tau(\omega) &= 0 && \begin{cases} 0.25 < \omega < 0.7 \end{cases}
\end{aligned} \tag{7.1}$$

$$\eta_d(\omega) = \begin{cases} 0 \text{ dB} & 0 \leq \omega \leq 0.25 \text{ e } 0.7 \leq \omega \leq \pi \\ -30 \text{ dB} & 0.3 \leq \omega \leq 0.65 \end{cases}$$

$$\tau_d(\omega) = \begin{cases} 0 \text{ amostras} & 0 \leq \omega \leq 0.25 \text{ e } 0.7 \leq \omega \leq \pi \end{cases}$$

$$N = 8 \quad p = 2 \quad q = 2 \quad n = 10$$

onde na conjunto de especificações em (7.1),  $N$  é a ordem do filtro FIR,  $p$  é o valor da norma  $L_p$  de aproximação da magnitude do filtro,  $q$  é o valor da norma  $L_q$  de aproximação do atraso de grupo do filtro e  $n$  é a dimensão do vetor de pesos  $\mathbf{W}$  do método ponderado de escalarização da função multi-objetivo. Note que na especificação da resposta desejada do atraso de grupo  $\tau_d(\omega)$  o valor de 0 amostras é utilizado. Embora esse requisito não seja fisicamente realizável com um número finito de coeficientes do filtro FIR, o valor força a otimização a encontrar o menor valor possível do atraso de grupo na banda de frequências desejadas.

A Figura 7.5 apresenta o gráfico da magnitude da resposta em frequência e atraso de grupo desejado para as especificações em (7.1).

O número de avaliações das funções objetivos  $f_1$ , equação (3.31) e  $f_2$ , equação (3.32) na análise exploratória estatística foram restringidas à 20000 avaliações para cada minimização (realização) do algoritmo de otimização em estudo. Essa restrição foi imposta para acelerar a geração dos resultados da análise devido ao alto custo computacional e tempo de execução dos experimentos estatísticos.

Adicionalmente, para cada experimento estatístico, o vetor de referência  $\mathbf{Z}^*$  para o cálculo do indicador  $\epsilon$ -binário  $I_\epsilon(\mathbf{Z}^*, \mathbf{A})$  de cada observação é rigorosamente o mesmo.

Tabela 7.1: Finalização dos intervalos para o algoritmo genético

Parâmetro	Intervalo
Crossover	0.7 – 1
Mutação	0.1 – 0.5

### 7.3.1 Resultados da Estimação dos Parâmetros Ótimos dos Algoritmos

#### 7.3.1.1 Resultados para o algoritmo genético

Seguindo os passos apresentados na metodologia estatística da subseção 6.4.1 para estimação dos valores de parâmetros ótimos para a taxa de mutação  $p_{mu}$  e a taxa de *crossover*  $p_{cr}$  do algoritmo genético, primeiramente, foi criado um experimento em blocos completamente aleatorizados com 10 réplicas e valores de parâmetros  $0 \leq p_{mu} \leq 1$  e  $0 \leq p_{cr} \leq 1$  em intervalos de valor igual a 0.1 para discretização dos valores. Uma instância de censura foi identificada nesse experimento quando a diferença entre o indicador  $\epsilon$ -binário do melhor indivíduo  $I_{\epsilon}(\mathbf{Z}^*, \mathbf{A}_m)$  e o indicador  $\epsilon$ -binário do pior indivíduo  $I_{\epsilon}(\mathbf{Z}^*, \mathbf{A}_p)$  na observação foi menor ou igual a 0.94. O diagrama de pontos para esse limiar pode ser visto na Figura 7.6.

Com relação ao valor limiar utilizado no experimento, o mesmo foi escolhido para ajustar o diagrama de pontos à uma configuração desejada. Se, por exemplo, o diagrama de pontos para um determinado valor limiar possuir pontos em todos os valores dos parâmetros, então isso significa que o limiar é muito alto, ou seja, o limiar nunca é alcançado. Dessa forma, o limiar é então decrementado, por exemplo, de 0.9 para 0.8, o qual resulta em algumas áreas do diagrama sem pontos. Se existir para um certo valor limiar, um espaço sem pontos no diagrama indicando um número de níveis que possam ser testados, então esse limiar se tornou o valor escolhido para geração do diagrama. Esse processo pode ser visto graficamente na Figura 7.7 para os valores limiares 1, 0.99 e 0.94, onde a cada redução do limiar um maior número de níveis para teste são gerados.

Dando prosseguimento à metodologia, o próximo passo consiste na finalização dos intervalos iniciais dos parâmetros. Para isso, 10 experimentos de 100 réplicas cada foram gerados usando os valores não censurados dos parâmetros no diagrama de pontos da Figura 7.6 como níveis de teste dos experimentos. A combinação de parâmetros resultando na melhor performance, ou seja, com o maior valor de  $I_{\epsilon}(\mathbf{Z}^*, \mathbf{A})$ , foi achada em cada experimento. Quando esses valores são agrupados, eles demonstram os níveis inferiores e superiores dos parâmetros associados com a melhor performance. A tabela 7.1 apresenta os intervalos resultantes para esse processo.

Após obter os intervalos finais de valores dos parâmetros, um experimento em blocos completamente aleatorizados de 100 réplicas foi gerado com os valores de parâmetros da tabela 7.1. A tabela ANOVA para esse experimento é ilustrada em 7.2.

Na tabela ANOVA em 7.2, a coluna GL representa o grau de liberdade associado com cada

Tabela 7.2: Tabela ANOVA com somatório particionado dos quadrados do experimento com algoritmo genético

Parâmetro	GL	SQ	MQ	Valor $F$	Pr (F)
Crossover	3	0.00081081	0.0002702687	15.94334	0.0000000
Potência de 1	1	0.00017910	0.0001790981	10.56512	0.0011735
Potência de 3	1	0.00053700	0.0005370030	31.67817	0.0000000
Mutação	4	0.00096178	0.0002404441	14.18396	0.0000000
Potência de 2	1	0.00024277	0.0002427656	14.32091	0.0001591
Potência de 3	1	0.00046401	0.0004640082	27.37216	0.0000002
Interação	12	0.00145158	0.0001209646	7.13579	0.0000000
Potência2:Potência1	1	0.00021802	0.0002180192	12.86110	0.0003443
Potência3:Potência3	1	0.00017644	0.0001764378	10.40819	0.0012769

parâmetro, SQ representa o somatório dos quadrados e MQ a média quadrática para cada parâmetro.

Adicionalmente, conforme visto na tabela 7.2, o parâmetro interação é altamente significativo  $\text{Pr}(F) = 0.0000000$ , dessa forma, não é necessário aumentar o tamanho amostral (número de réplicas do experimento).

Ainda com relação aos resultados da ANOVA, na tabela 7.2 é informado o somatório particionado dos quadrados para cada fator (mutação e *crossover*). Esses valores são necessários para avaliar o grau dos polinômios usados na regressão linear para geração da curva/superfície de resposta do experimento. Os valores estatisticamente significativos do somatório particionado foram selecionados para a equação da superfície de resposta ajustada do experimento:

$$\begin{aligned}
 S_{\text{ga}} = & -0.5128 + 1.7206x_{\text{cr}} + 0.8005x_{\text{cr}}^3 + 0.0904x_{\text{mu}}^2 - 0.0915x_{\text{mu}}^3 \\
 & -2.0377x_{\text{cr}}^2 - 0.0185x_{\text{mu}} - 0.0072x_{\text{cr}}^2x_{\text{mu}} - 0.0069x_{\text{cr}}^3x_{\text{mu}}^3
 \end{aligned}
 \tag{7.2}$$

O gráfico em 3 dimensões da superfície de resposta (7.2) do experimento pode ser visto na Figura 7.8.

Finalmente, para obter a estimativa dos valores ótimos dos parâmetros do experimento, o ponto máximo dessa superfície deve ser encontrado. Esse problema foi postulado como um problema de otimização com restrição por:

Tabela 7.3: Finalização dos intervalos para a otimização por enxame de partículas

Parâmetro	Intervalo
Parâmetro de controle $\phi$	2 – 3.6
Momento de inércia $\alpha$	0.6 – 0.8

$$\begin{aligned}
 &\max_x S_{\text{ga}}(x_{\text{mu}}, x_{\text{cr}}) \\
 &\text{sujeito a} \\
 &0.7 \leq x_{\text{cr}} \leq 1 \\
 &0.1 \leq x_{\text{mu}} \leq 0.5
 \end{aligned} \tag{7.3}$$

Embora o problema em (7.3) seja não-linear, particularmente para funções polinomiais, como por exemplo a função da superfície de resposta em (7.2), algoritmos para otimização global de funções polinomiais são derivados através de uma técnica que envolve o uso do somatório dos quadrados e programação semi-definida [70]. Com essa técnica, os valores globais para o problema de otimização em (7.3) obtidos com a biblioteca MATLAB em [71] foram:

$$x_{\text{cr}}^* = 0.7701 \quad x_{\text{mu}}^* = 0.4639 \tag{7.4}$$

O valor da superfície de resposta para os valores ótimos em (7.4) é:

$$S_{\text{ga}}(x_{\text{mu}}^*, x_{\text{cr}}^*) = 0.9944 \tag{7.5}$$

### 7.3.1.2 Resultados para a otimização por enxame de partículas

Seguindo a metodologia da subseção 7.3.1.1 com a finalidade de obtenção da estimativa dos parâmetros ótimos para o algoritmo OEA, o primeiro passo consiste na geração de um experimento em blocos completamente aleatorizados com 10 réplicas e valores de parâmetros  $0 \leq \alpha < 1$  e  $0 < \phi < 4$ . O valor de limiar selecionado para o experimento foi de 0.9. A Figura 7.9 apresenta o diagrama de pontos resultante.

Dando prosseguimento à metodologia, o próximo passo consiste na finalização dos intervalos iniciais dos parâmetros. Para isso, 10 experimentos de 100 réplicas cada foram gerados usando o valores não censurados dos parâmetros no diagrama de pontos da Figura 7.9 como níveis de teste dos experimentos. A combinação de parâmetros resultando na melhor performance foi achada em cada experimento. A tabela 7.3 apresenta os intervalos resultantes para esse processo.

Após obter os intervalos finais de valores dos parâmetros, um experimento em blocos completamente aleatorizados de 100 réplicas foi gerado com os valores de parâmetros da tabela 7.3. A tabela ANOVA para esse experimento é ilustrada em 7.4.



Tabela 7.4: Tabela ANOVA com somatório particionado dos quadrados do experimento com a otimização por enxame de partículas

Parâmetro	GL	SQ	MQ	Valor $F$	Pr (F)
Inércia $\alpha$	2	3.956673	1.978336	15.94334	0.000000
Potência de 1	1	1.080438	1.080438	1232.871	0.000000
Potência de 2	1	2.876234	2.876234	3282.025	0.000000
Controle $\phi$	8	4.045458	0.505682	577.026	0.000000
Potência de 1	1	0.315420	0.315420	359.921	0.000000
Potência de 2	1	0.364443	0.364443	415.861	0.000000
Interação	16	6.585366	0.411585	469.653	0.000000
Potência1:Potência1	1	0.058580	0.058580	66.844	$4.440892 \times 10^{-16}$
Potência2:Potência1	1	0.505633	0.505633	576.969	0.000000
Potência1:Potência2	1	0.105872	0.105872	120.809	0.000000
Potência2:Potência2	1	0.434902	0.434902	496.260	0.000000

Novamente, conforme visto na tabela 7.4, o parâmetro interação é altamente significativo  $\text{Pr}(F) = 0.000000$ , dessa forma, não é necessário aumentar o tamanho amostral (número de réplicas do experimento).

Os valores estatisticamente significativos do somatório particionado foram selecionados para a equação da superfície de resposta ajustada do experimento:

$$\begin{aligned}
 S_{\text{oea}} = & \\
 & 6.3771 - 15.5283x_{\alpha} + 11.0608x_{\alpha}^2 - 5.1732x_{\phi} + 0.8695x_{\phi}^2 \\
 & + 15.1011x_{\alpha}x_{\phi} - 10.9736x_{\alpha}^2x_{\phi} - 2.5361x_{\alpha}x_{\phi}^2 + 1.8406x_{\alpha}^2x_{\phi}^2
 \end{aligned} \tag{7.6}$$

O gráfico em 3 dimensões da superfície de resposta (7.6) do experimento pode ser visto na Figura 7.10.

Finalmente, para obter a estimativa dos valores ótimos dos parâmetros do experimento, o ponto máximo dessa superfície deve ser encontrado. Esse problema foi postulado como um problema de otimização com restrição por:

$$\begin{aligned}
 & \max_x S_{\text{oea}}(x_{\phi}, x_{\alpha}) \\
 & \text{sujeito a} \\
 & 2 \leq x_{\phi} \leq 3.6 \\
 & 0.6 \leq x_{\alpha} \leq 0.8
 \end{aligned} \tag{7.7}$$

Os valores obtidos para o problema de otimização em (7.7) utilizando a biblioteca MATLAB

Tabela 7.5: Tabela ANOVA para o experimento de comparação dos algoritmos

Parâmetro	GL	SQ	MQ	Valor $F$	Pr (F)
Algoritmo de Otimização	3	0.05164	0.01721	1.93	0.01

Tabela 7.6: Estatísticas de  $I_\epsilon(Z^*, A)$  para o experimento comparativo de performance

Algoritmo	Máximo	Mínimo	Média	Variância
Algoritmo Genético	0.9959	0.9854	0.9921	$7.7668 \times 10^{-6}$
Otimização por Enxame de Partículas	0.9823	0.9360	0.9600	$2.0947 \times 10^{-4}$
Quasi-Newton	0.9968	0.5472	0.9040	0.0354
Resfriamento Simulado	0.9964	0.9891	0.9918	$4.1675 \times 10^{-6}$

em [71] foram:

$$x_\phi^* = 2.5448 \quad x_\alpha^* = 0.6550 \quad (7.8)$$

O valor da superfície de resposta para os valores ótimos em (7.8) é:

$$S_{\text{oea}}(x_\phi^*, x_\alpha^*) = 0.9642 \quad (7.9)$$

### 7.3.2 Resultados da Comparação de Performance dos Algoritmos de Otimização

Seguindo a metodologia estatística da subseção 6.4.2, para a comparação efetiva entre os algoritmos de otimização propostos no presente trabalho aplicados no projeto de um filtro FIR rejeita-faixa com especificações dadas em (7.1), criou-se um experimento em blocos completamente aleatorizados de 100 réplicas com os quatro algoritmos de otimização propostos no trabalho, Quasi-Newton, algoritmo genético, resfriamento simulado e otimização por enxame de partículas. Os resultados da ANOVA do experimento estão listados na tabela 7.5.

Conforme visto na tabela 7.5, a variação do parâmetro algoritmo de otimização possui alta significância estatística na resposta do experimento (valor do indicador  $\epsilon$ -binário).

Seguindo a metodologia, a tabela 7.6 apresenta os momentos estatísticos do indicador  $\epsilon$ -binário para o experimento comparativo.

Conforme visto na tabela 7.6, o algoritmo que apresentou melhor performance, na média, foi o algoritmo genético, seguido do resfriamento simulado, otimização por enxame de partículas e por fim, Quasi-Newton, o algoritmo com pior performance no experimento.

## 7.4 CONSIDERAÇÕES ADICIONAIS EM PROJETOS DE FILTROS COM A METODOLOGIA DE OTIMIZAÇÃO MULTI-OBJETIVO

### 7.4.1 Projeto de filtro FIR Rejeita-Faixa com Atraso de Grupo Reduzido nas Bandas de Passagem

Nessa seção, os algoritmos de otimização propostos foram utilizados na otimização do projeto de um filtro FIR rejeita-faixa com especificações dadas na equação (7.1). A figura 7.11 apresenta o conjunto Pareto-ótimo encontrado por cada algoritmo de otimização em uma típica realização.

Na Figura 7.11 também é informado o valor do indicador  $\epsilon$ -binário para cada conjunto Pareto. Adicionalmente, observa-se na figura que o algoritmo OEA obteve o filtro com o melhor valor para a distância de aproximação do atraso de grupo, conforme visto no ponto extremo do conjunto Pareto com distância de aproximação do atraso de grupo de  $\simeq 1.35 \times 10^5$ , além do filtro com o melhor valor de aproximação para a distância de aproximação da magnitude da resposta em frequência, junto com o algoritmo genético, conforme visto nos dois pontos extremos dos conjuntos Pareto-ótimos com distância de aproximação da magnitude da resposta em frequência de  $\simeq 6.02 \times 10^6$ . Embora os melhores valores extremos tenham sido alcançados com o algoritmo OEA, alguns pontos de seu conjunto Pareto possuem valores bastante elevados da distância de aproximação da magnitude e atraso de grupo, e por esse motivo, considerando o conjunto Pareto como todo, obteve-se o pior valor do indicador  $\epsilon$ -binário com esse algoritmo. O algoritmo Quasi-Newton obteve o melhor valor do indicador  $\epsilon$ -binário nessa realização, e o formato do conjunto Pareto-ótimo encontrado aparenta ser o de um conjunto convexo com forma quadrática.

As Figuras 7.12 e 7.13 apresentam, respectivamente, o comparativo da magnitude da resposta em frequência e do atraso de grupo para filtros do conjunto Pareto-ótimo com os algoritmos Quasi-Newton e otimização por enxame de partículas. Conforme pode ser visto nas figuras, observa-se claramente o compromisso entre a aproximação simultânea conflitante da magnitude e atraso de grupo do filtro. Se a distância de aproximação é reduzida para a magnitude, perde-se consequentemente na aproximação da distância do atraso de grupo do filtro e vice-versa. Convém notar ainda que as Figuras 7.12 e 7.13 mostram alguns pontos do conjunto Pareto de cada algoritmo através de sua coordenada  $x$  e  $y$  do gráfico da Figura 7.11. Por fim, convém notar que é observado de forma mais acentuada o conflito entre a aproximação simultânea no conjunto Pareto obtido pelo algoritmo otimização por enxame de partículas. Tome como exemplo o filtro com coordenada  $x = 139306$ ,  $y = 6062358$  na Figura 7.13. Esse filtro possui a melhor aproximação para a magnitude da resposta em frequência, por outro lado, no comparativo é o filtro que possui maior atraso de grupo nas bandas de passagem do filtro.

### 7.4.2 Projeto de filtro FIR Passa-Faixa com Atraso de Grupo Reduzido na Banda de Passagem

Nessa seção, os algoritmos de otimização propostos foram utilizados na otimização do projeto de um filtro FIR passa-faixa com as seguintes especificações:

$$\begin{aligned}
 W_{\eta}(\omega) &= 1 && \begin{cases} 0 \leq \omega \leq 0.25 \\ 0.3 \leq \omega \leq 0.65 \\ 0.7 \leq \omega \leq \pi \end{cases} \\
 W_{\eta}(\omega) &= 0 && \begin{cases} 0.25 < \omega < 0.3 \\ 0.65 < \omega < 0.7 \end{cases} \\
 W_{\tau}(\omega) &= 1 && \begin{cases} 0.3 \leq \omega \leq 0.65 \end{cases} \\
 W_{\tau}(\omega) &= 0 && \begin{cases} 0 \leq \omega < 0.3 \\ 0.65 < \omega \leq \pi \end{cases} \\
 \eta_d(\omega) &= \begin{cases} -30 \text{ dB} & 0 \leq \omega \leq 0.25 \text{ e } 0.7 \leq \omega \leq \pi \\ 0 \text{ dB} & 0.3 \leq \omega \leq 0.65 \end{cases} \\
 \tau_d(\omega) &= \begin{cases} 0 \text{ amostras} & 0.3 \leq \omega \leq 0.65 \end{cases} \\
 N = 8 & \quad p = 2 \quad q = 2 \quad n = 10
 \end{aligned} \tag{7.10}$$

A Figura 7.14 apresenta o gráfico da magnitude da resposta em frequência e atraso de grupo desejado para as especificações em (7.10).

A figura 7.15, apresenta o conjunto Pareto-ótimo encontrado por cada algoritmo de otimização em uma típica realização, aplicado ao problema proposto.

Novamente, observa-se na Figura 7.15 que o algoritmo otimização por enxame de partículas obteve o filtro com o melhor valor para a distância de aproximação do atraso de grupo, conforme visto no ponto extremo do conjunto Pareto com distância de aproximação do atraso de grupo de  $\simeq 1.32 \times 10^5$ , além do filtro com o melhor valor de aproximação para a distância de aproximação da magnitude da resposta em frequência, junto com o algoritmo genético, conforme visto nos dois pontos extremos dos conjuntos Pareto-ótimos com distância de aproximação da magnitude da resposta em frequência de  $\simeq 6 \times 10^6$ . Novamente, embora os melhores valores extremos tenham sido alcançados com o algoritmo otimização por enxame de partículas, alguns pontos de seu conjunto Pareto possuem valores bastante elevados da distância de aproximação da magnitude e atraso de grupo, e por esse motivo, considerando o conjunto Pareto como todo, obteve-se o pior valor do indicador  $\epsilon$ -binário com esse algoritmo. O algoritmo Quasi-Newton obteve o melhor

valor do indicador  $\epsilon$ -binário nessa realização, e o formato do conjunto Pareto-ótimo aparenta ser o de um conjunto convexo com forma quadrática.

Por fim, as Figuras 7.16 e 7.17 apresentam, respectivamente, o comparativo da magnitude da resposta em frequência e do atraso de grupo para filtros do conjunto Pareto-ótimo com os algoritmo genético e o resfriamento simulado. Novamente, observa-se o compromisso entre a aproximação simultânea conflitante da magnitude e atraso de grupo dos filtros do conjunto Pareto encontrados.

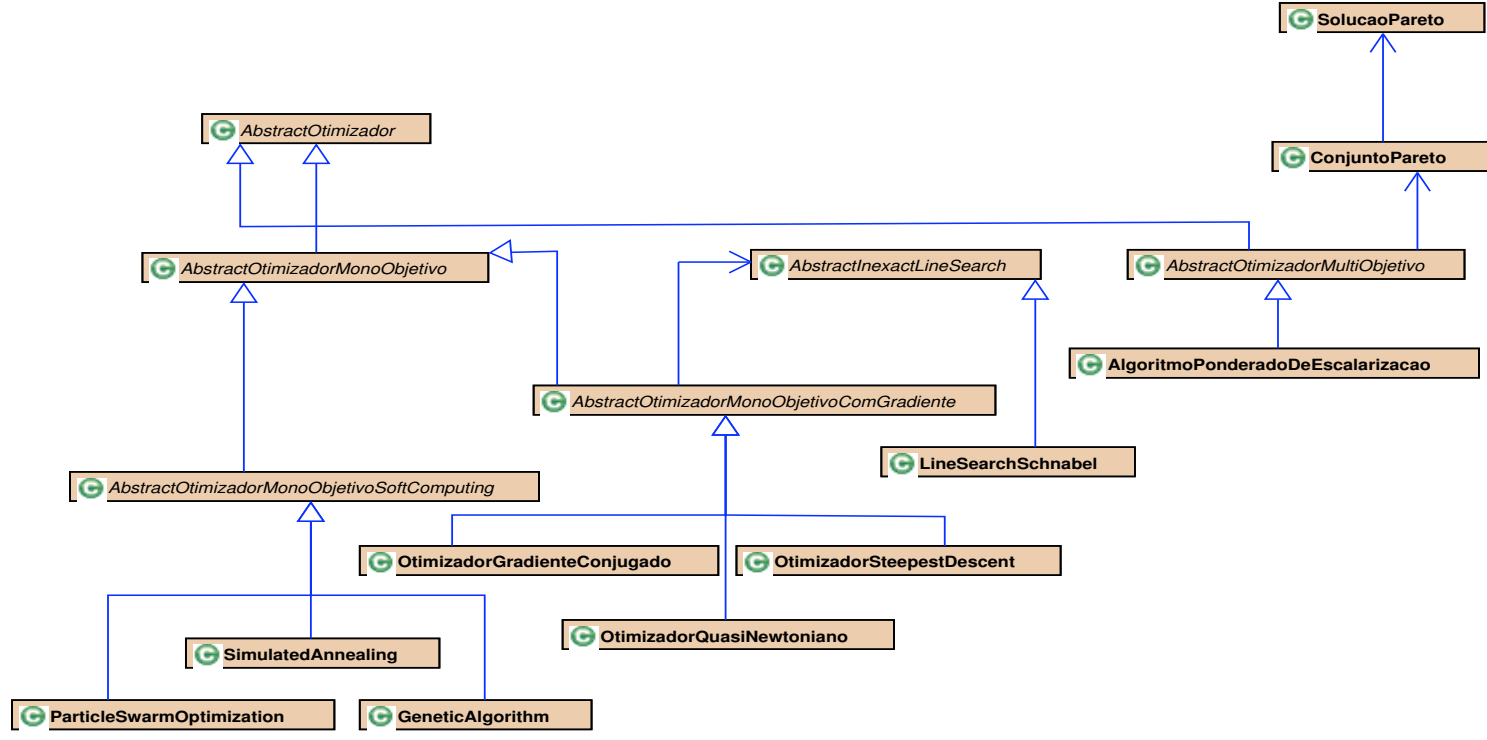


Figura 7.1: Diagrama de classes dos algoritmos de otimização

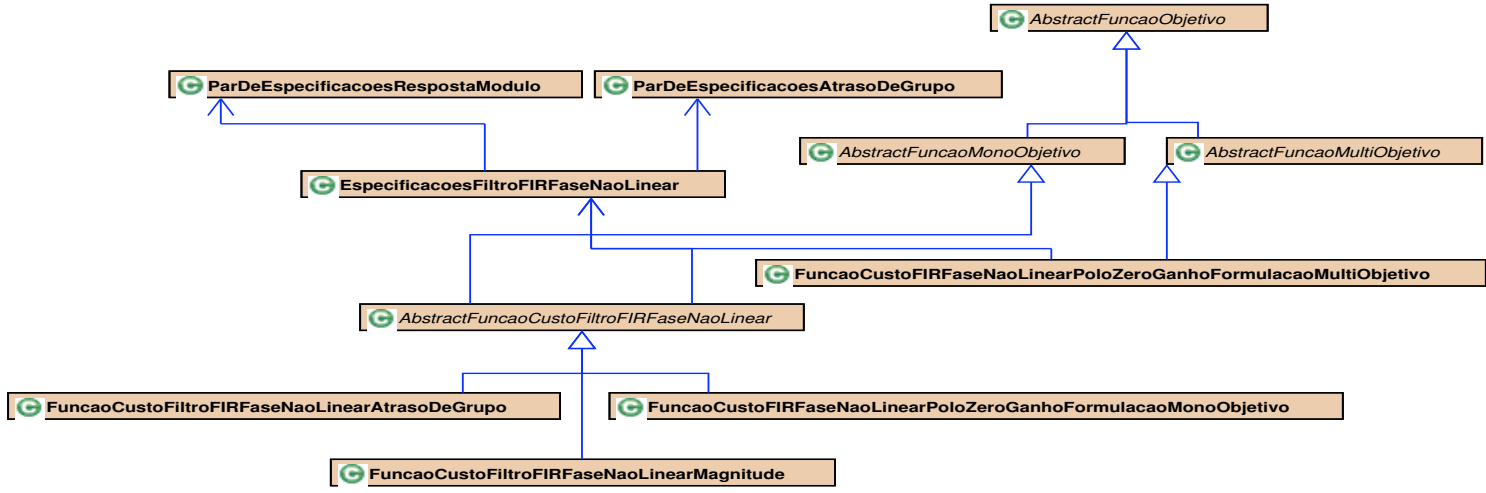


Figura 7.2: Diagrama de classes do projeto de filtros FIR

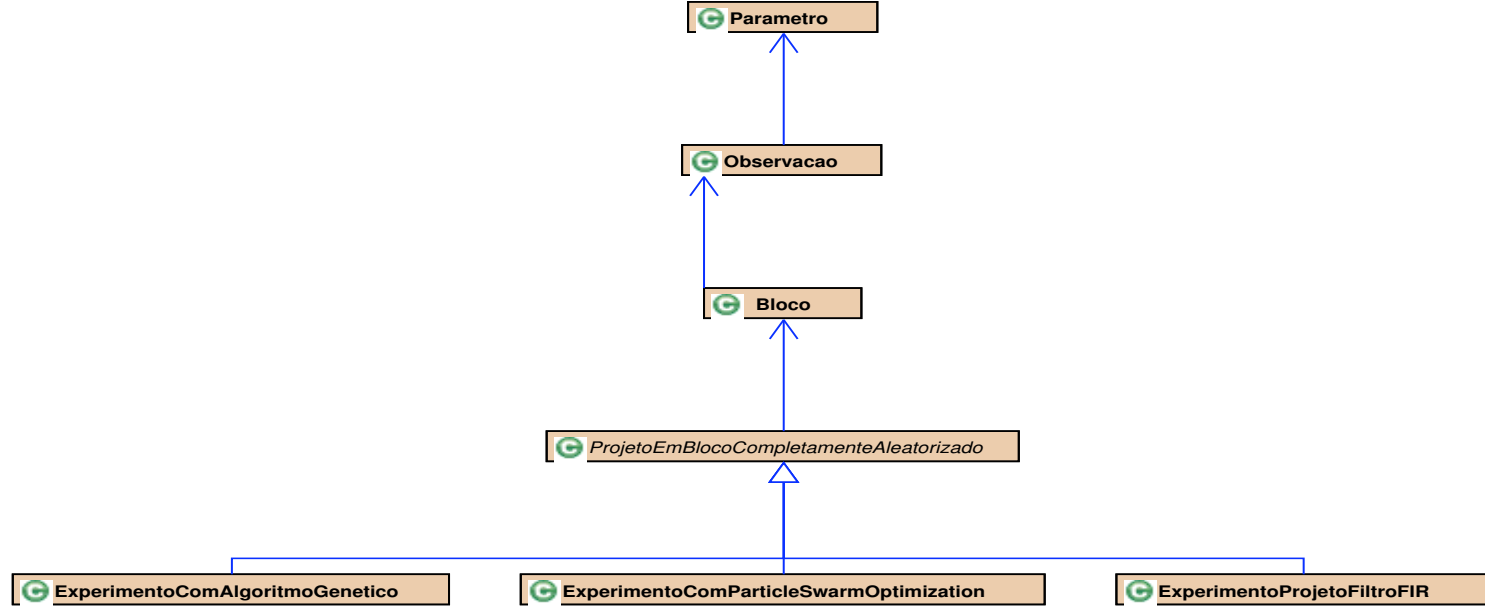
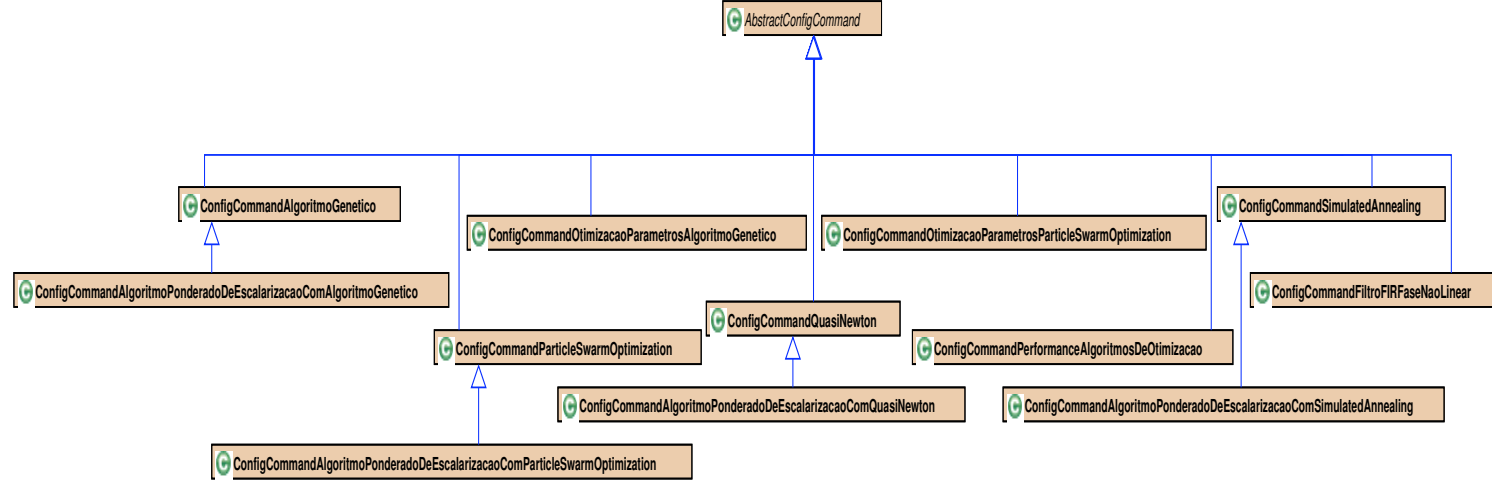
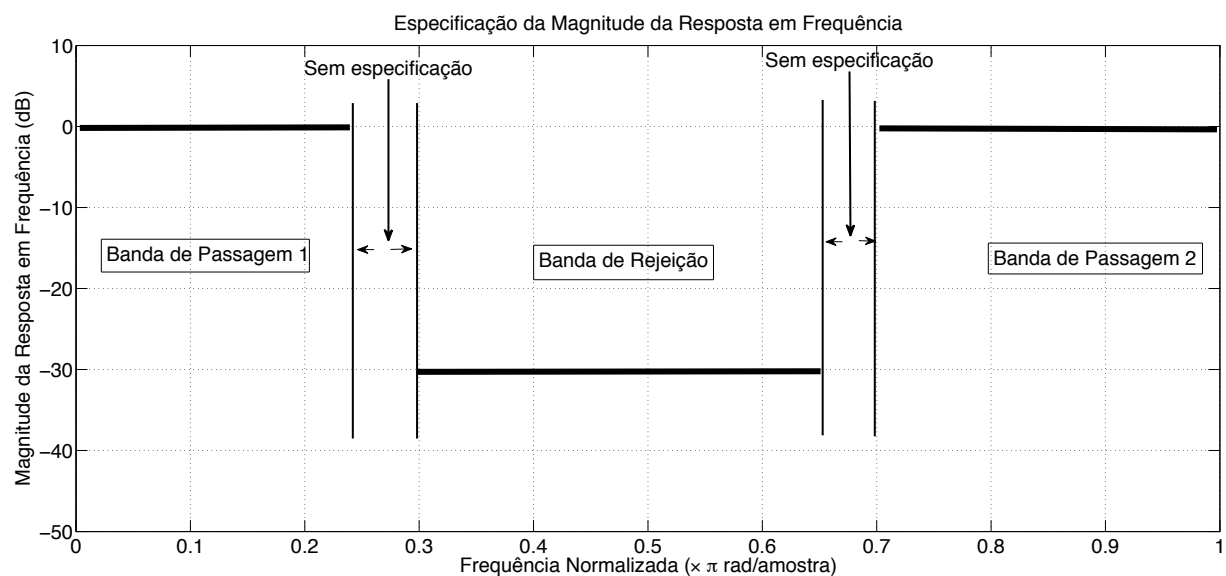


Figura 7.3: Diagrama de classes dos experimentos estatísticos

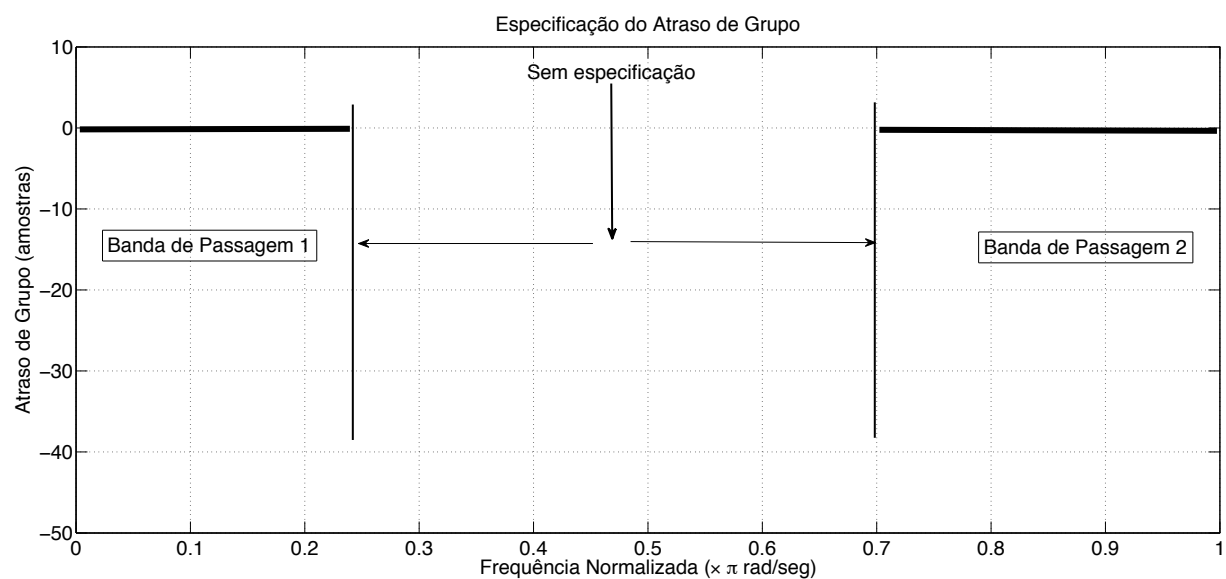


Figura 7.4: Diagrama de classes de configuração do framework de otimização





(a) Especificação da magnitude



(b) Especificação do atraso de grupo

Figura 7.5: Especificação da resposta em frequência do filtro rejeita-faixa para análise exploratória estatística

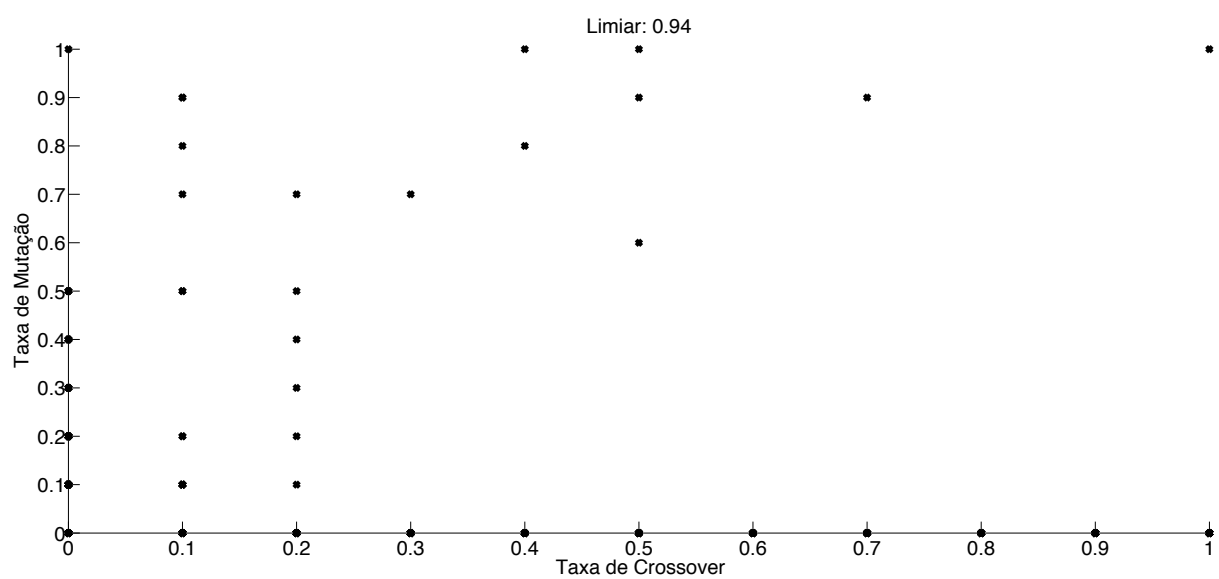
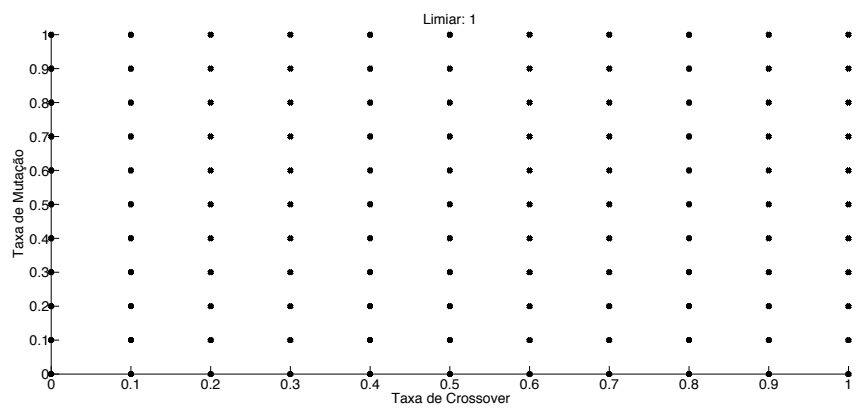
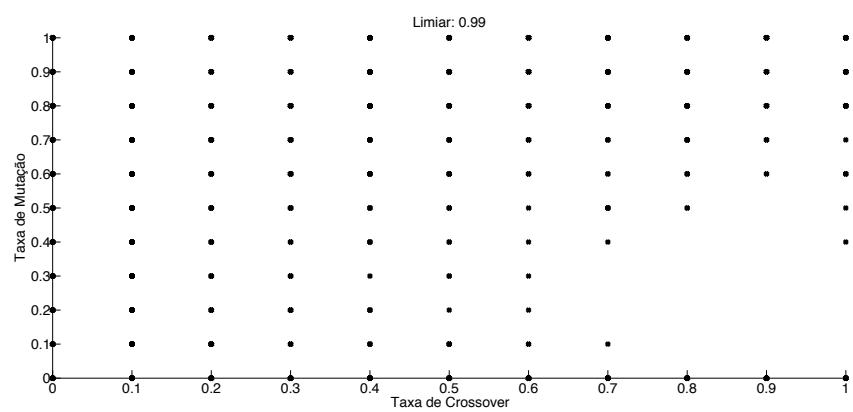


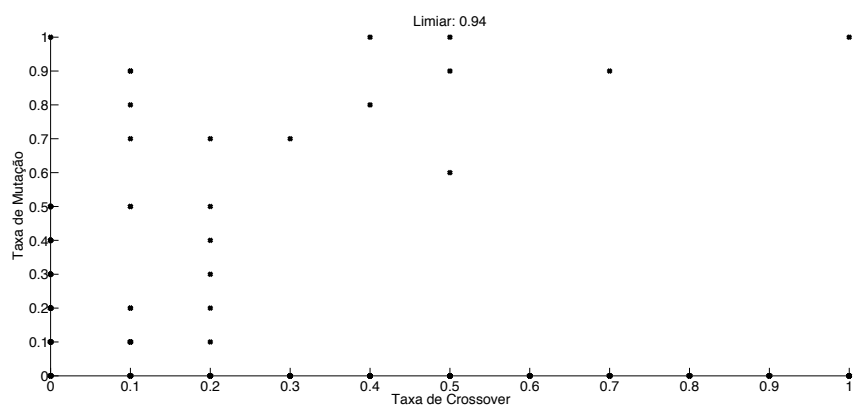
Figura 7.6: Diagrama de Pontos do algoritmo genético com limiar de 0.94



(a) Limiar 1



(b) Limiar 0.99



(c) Limiar 0.94

Figura 7.7: Processo de escolha do valor limiar no diagrama de pontos

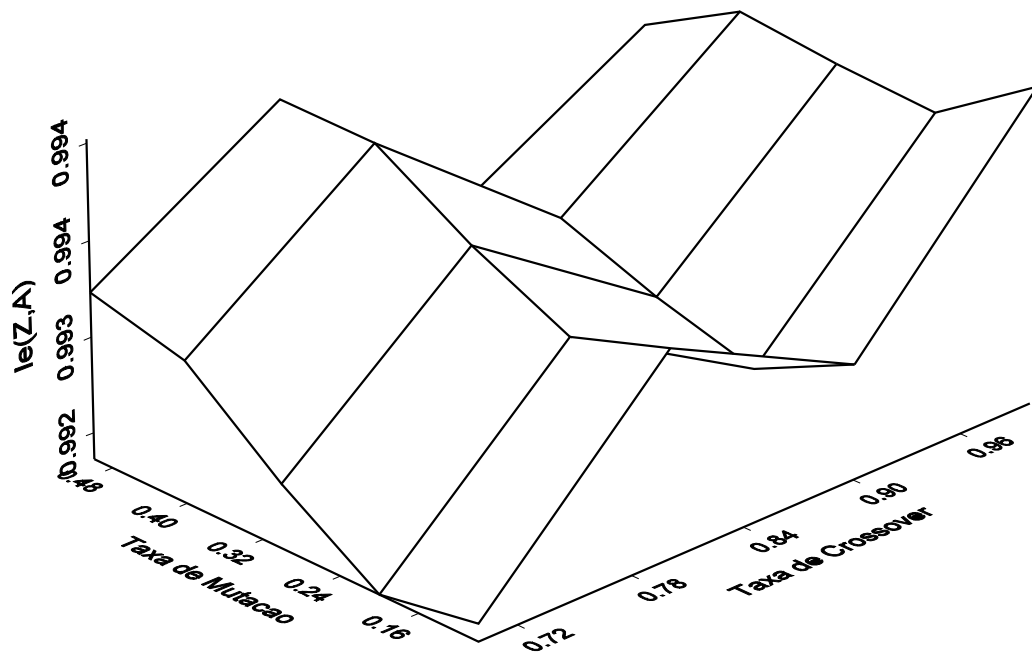


Figura 7.8: Superfície de resposta do experimento de estimação de parâmetros ótimos do algoritmo genético

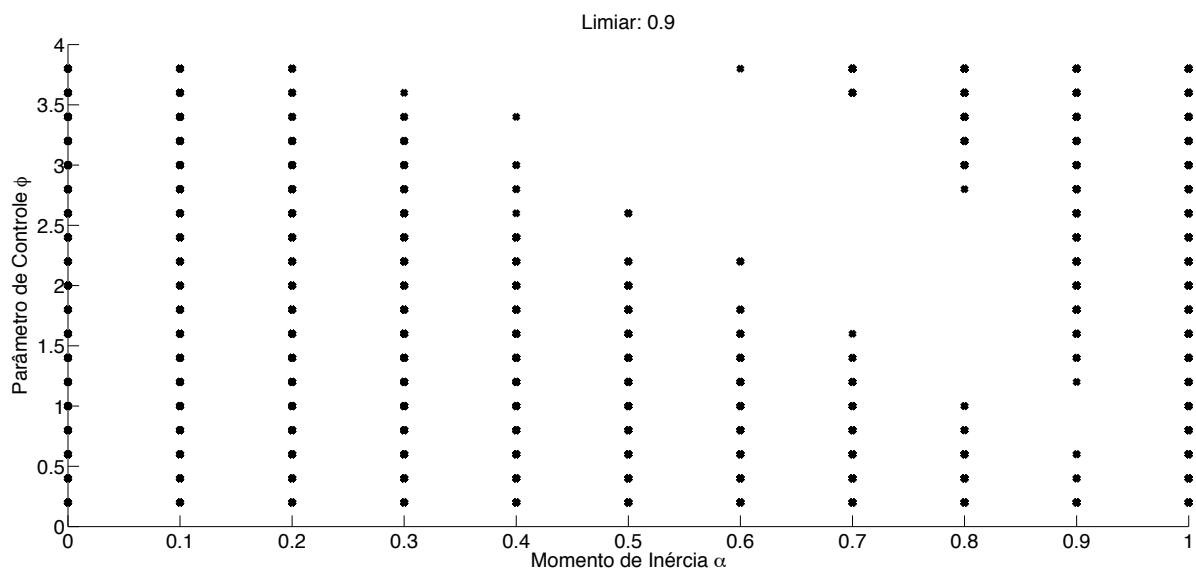


Figura 7.9: Diagrama de Pontos da otimização por enxame de partículas com limiar de 0.9

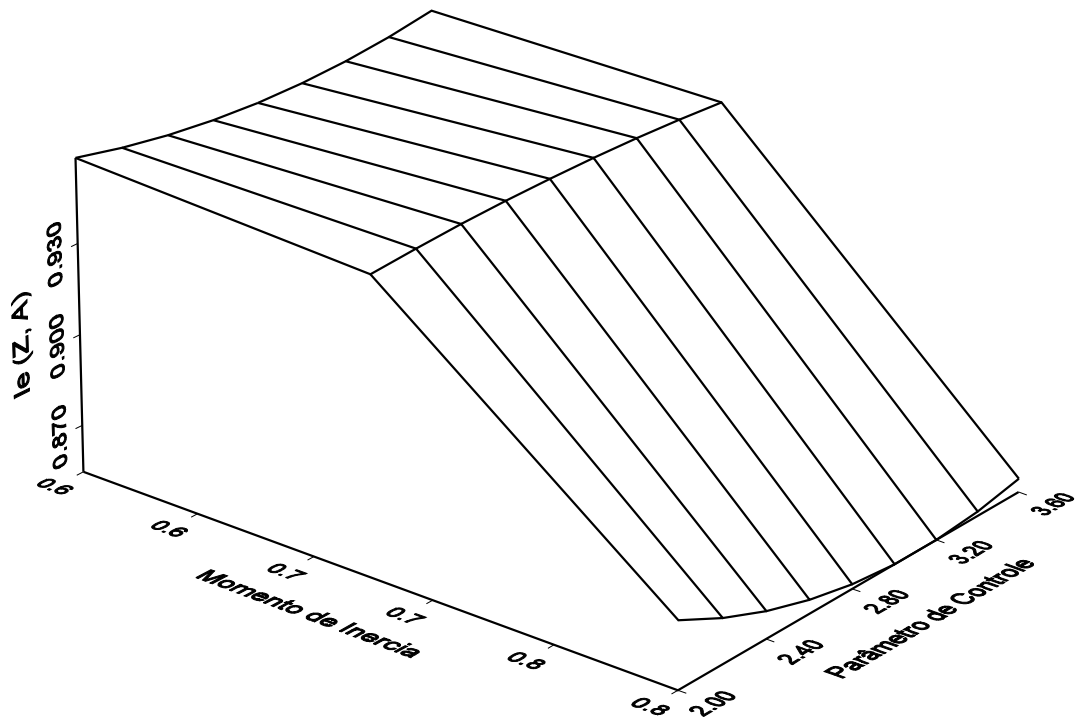


Figura 7.10: Superfície de resposta do experimento de estimação de parâmetros ótimos da otimização por enxame de partículas

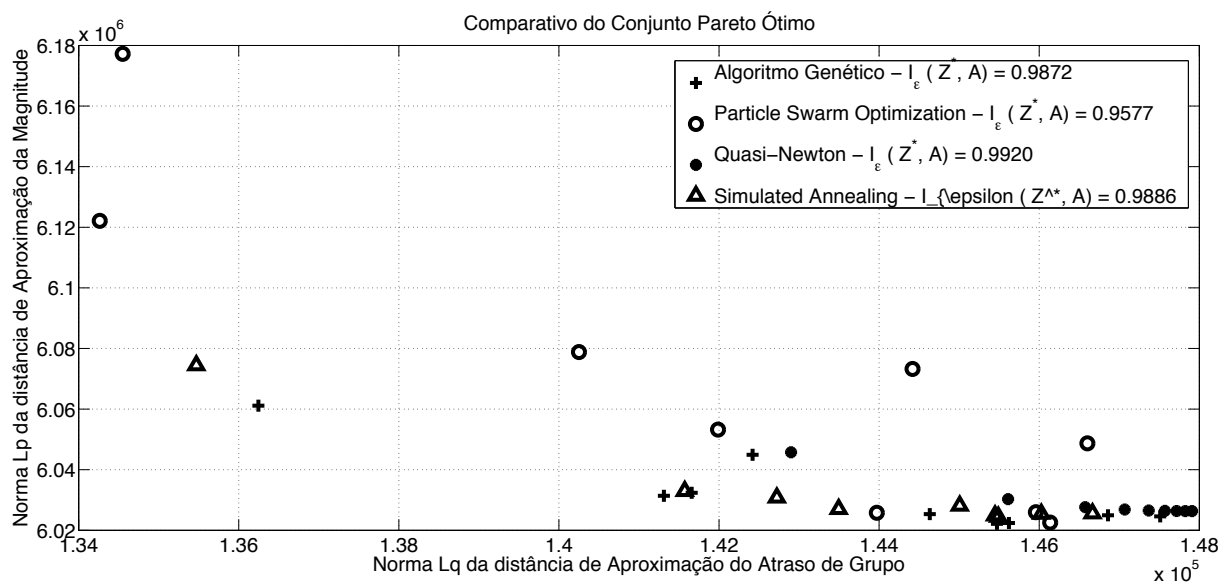
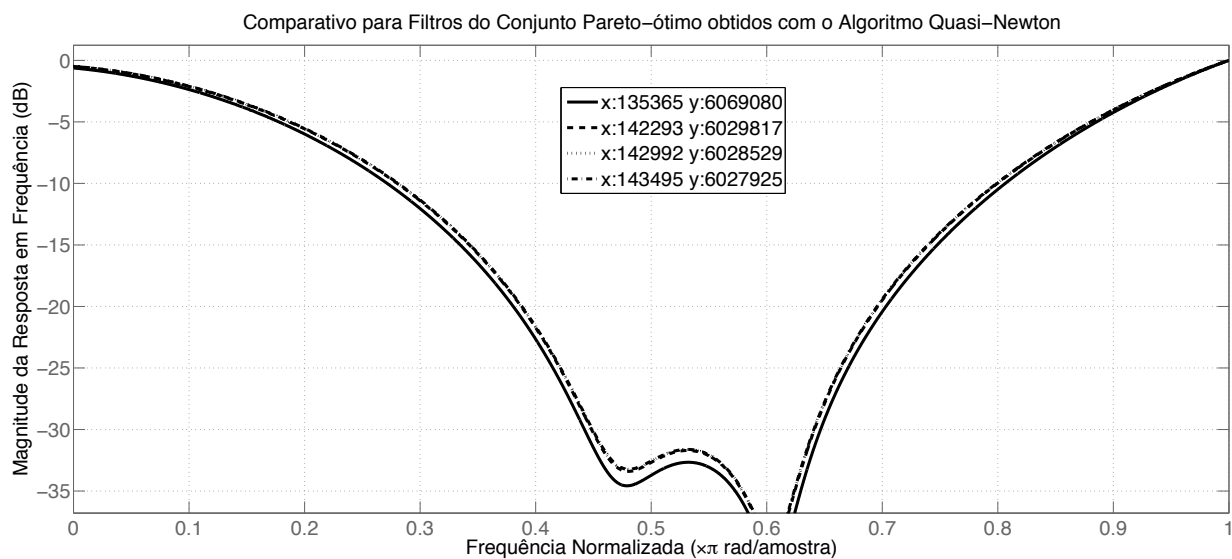
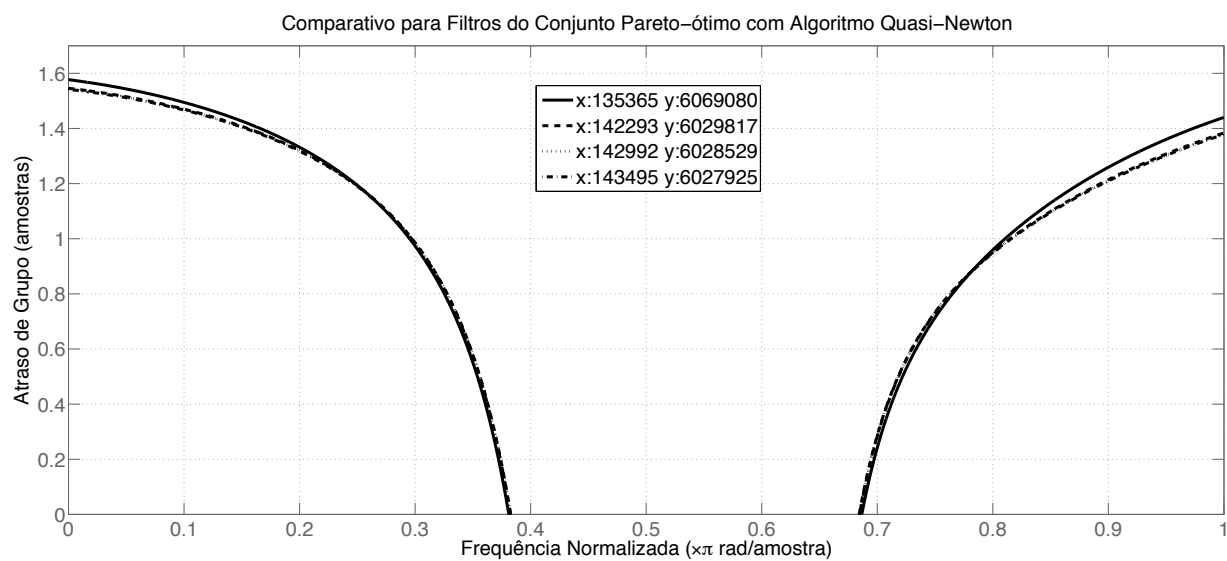


Figura 7.11: Comparativo do conjunto Pareto ótimo do projeto de filtro FIR rejeita-faixa

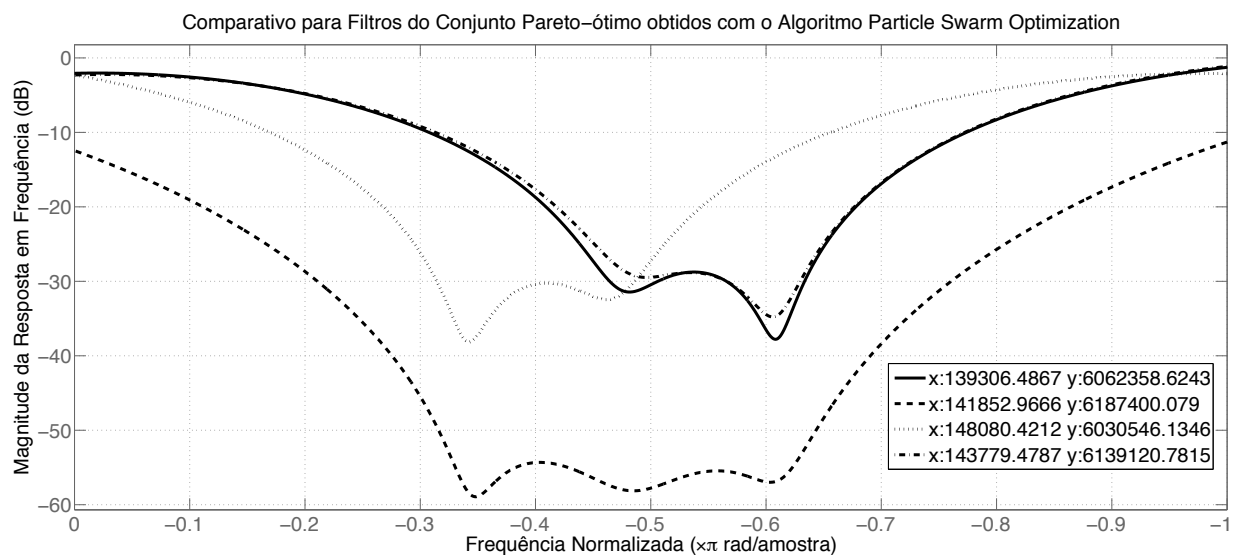


(a) Magnitude da Resposta em Frequência

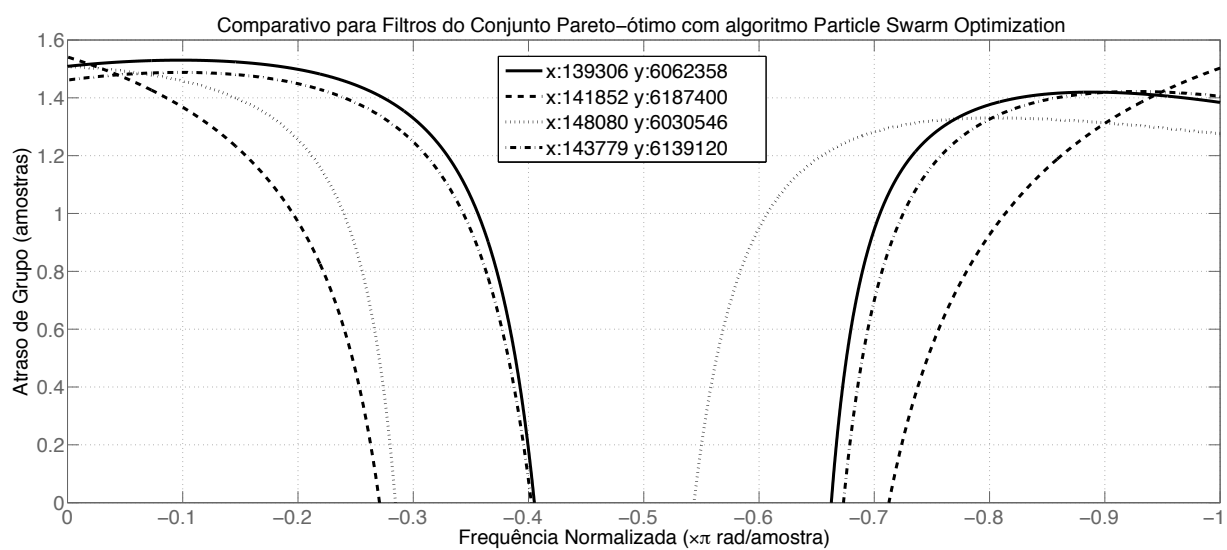


(b) Atraso de Grupo

Figura 7.12: Comparativo para diferentes filtros do conjunto Pareto-ótimo obtidos com o algoritmo Quasi-Newton



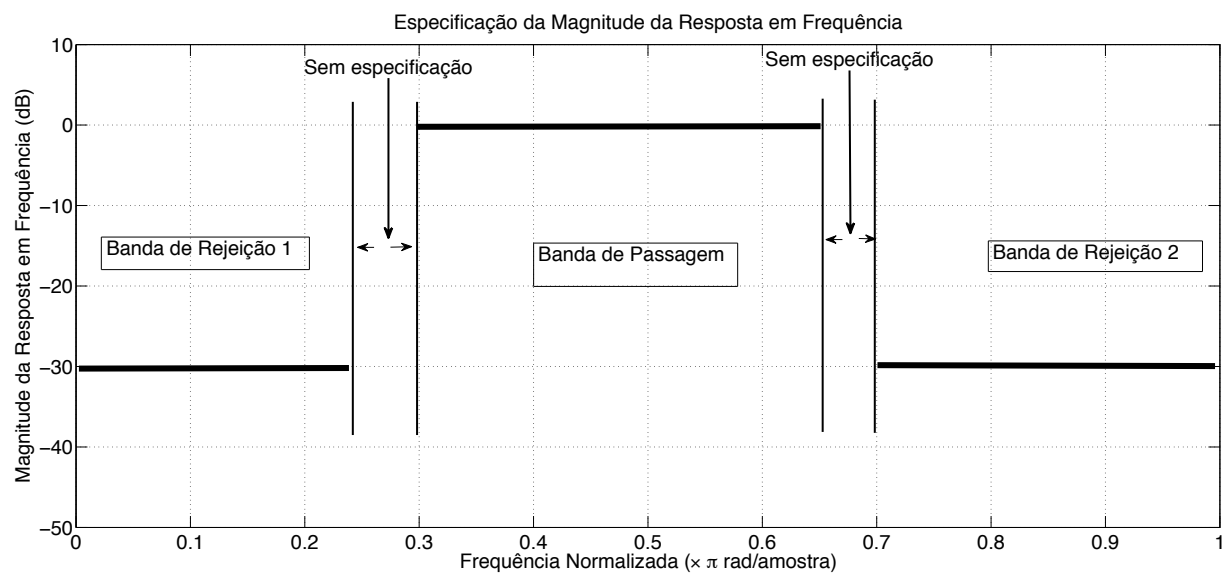
(a) Magnitude da Resposta em Frequência



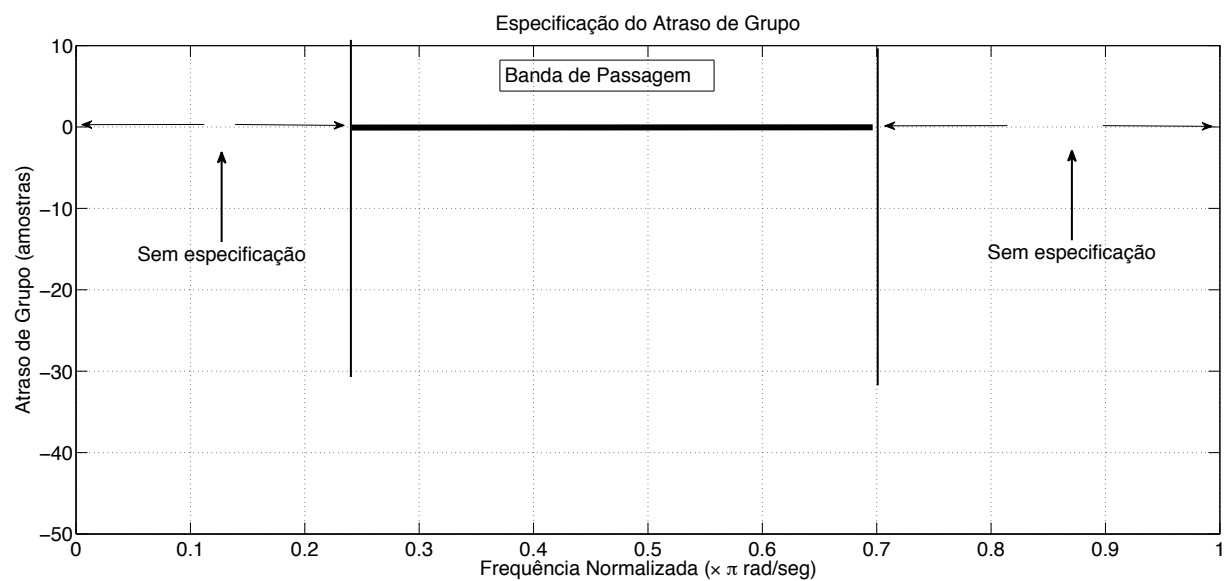
(b) Atraso de Grupo

Figura 7.13: Comparativo para diferentes filtros do conjunto Pareto-ótimo obtidos com o algoritmo otimização por enxame de partículas





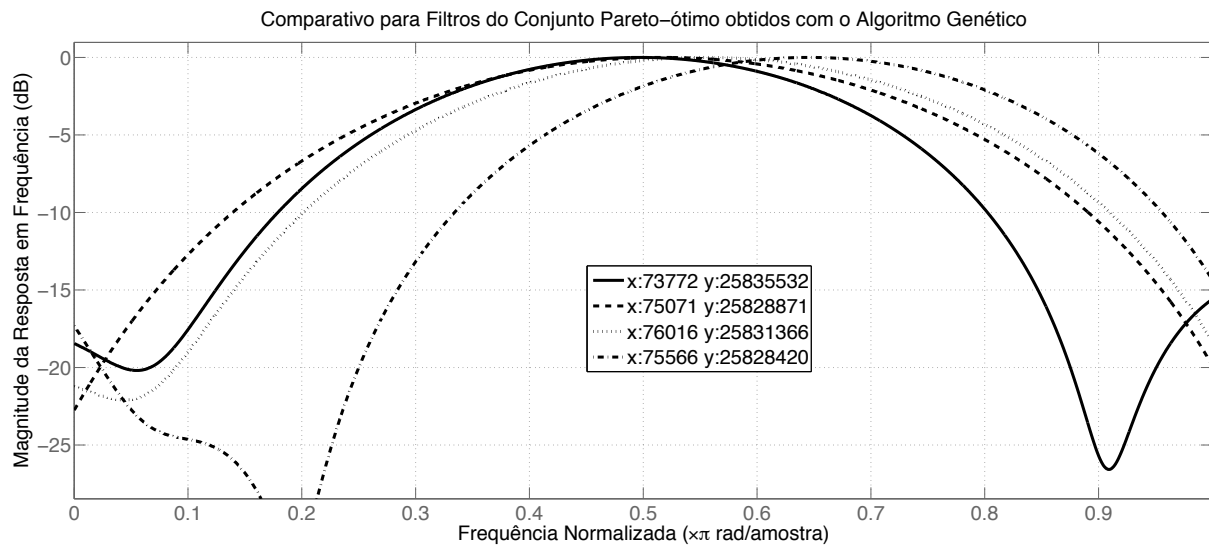
(a) Especificação da magnitude



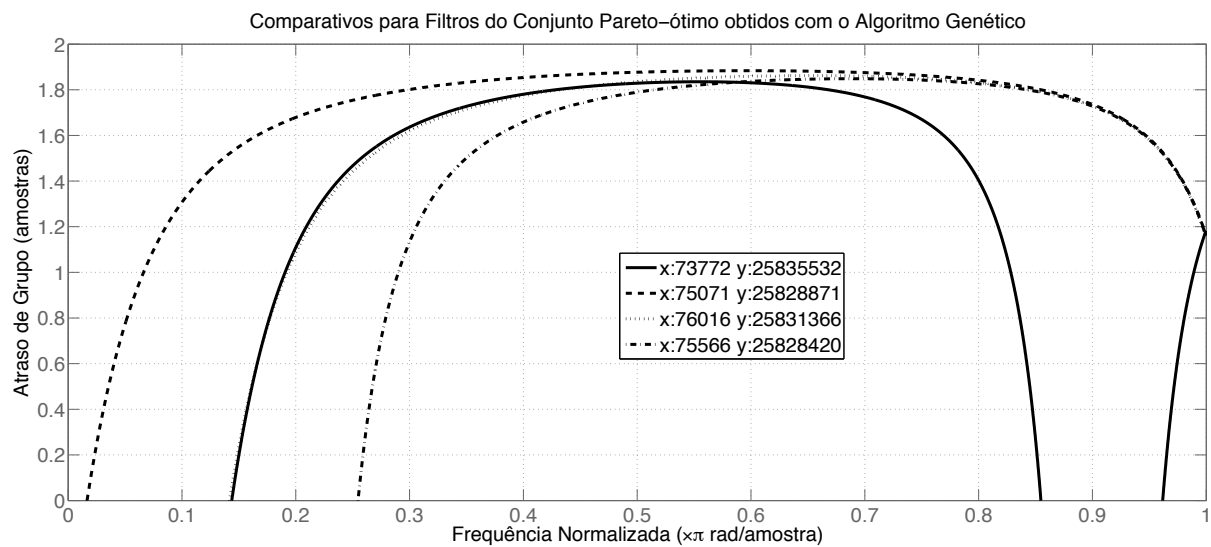
(b) Especificação do atraso de grupo

Figura 7.14: Especificação da resposta em frequência do filtro passa-faixa



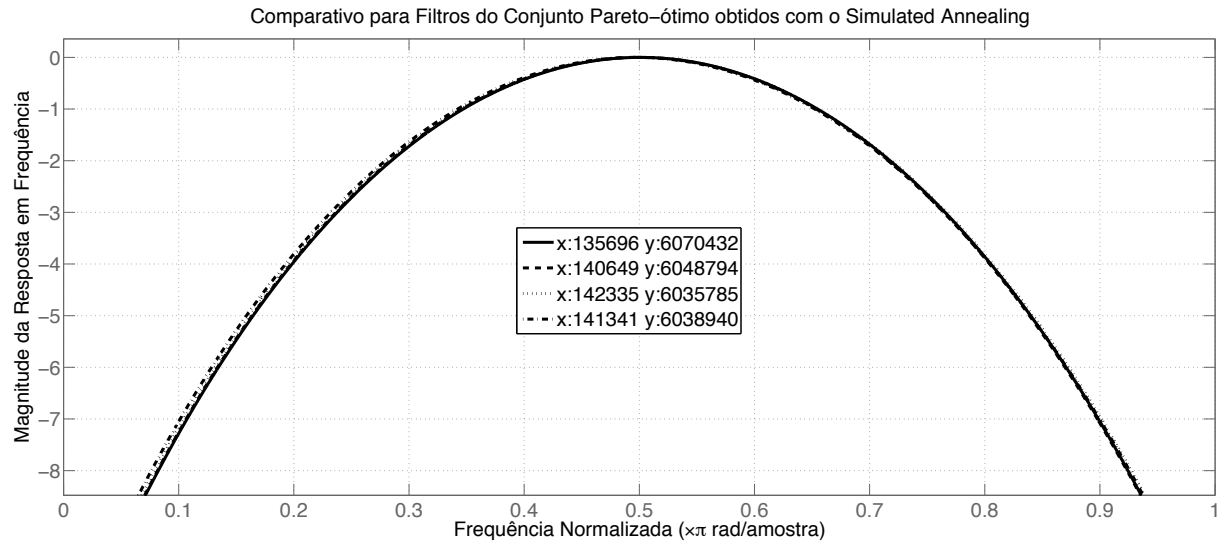


(a) Magnitude da Resposta em Frequência

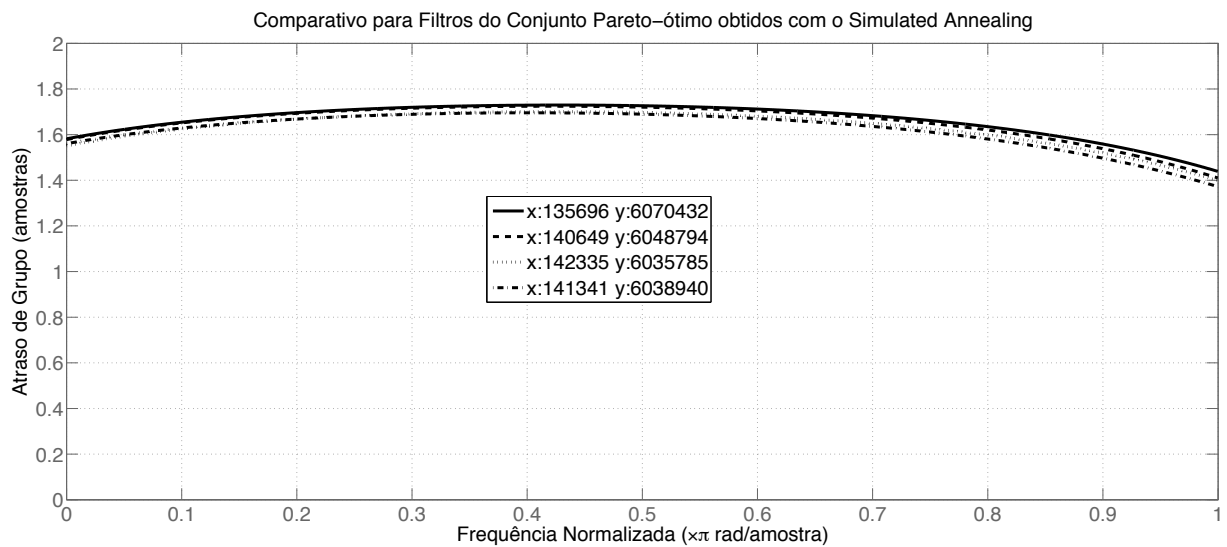


(b) Atraso de Grupo

Figura 7.16: Comparativo para diferentes filtros do conjunto Pareto-ótimo obtidos com o algoritmo genético



(a) Magnitude da Resposta em Frequência



(b) Atraso de Grupo

Figura 7.17: Comparativo para diferentes filtros do conjunto Pareto-ótimo obtidos com o resfriamento simulado



## 8 CONCLUSÕES

Concluindo, o presente trabalho teve por objetivo apresentar discussão detalhada do projeto ótimo de filtros digitais FIR de fase não-linear. Esse problema particular foi analisado sobre a óptica da teoria de otimização. O algoritmo de otimização quasi-Newton, baseado no gradiente da função objetivo, além de vários algoritmos de otimização *soft-computing*, foram propostos para resolução do problema multi-objetivo não-linear de projeto ótimo.

Como contribuição adicional, foi desenvolvido no presente trabalho uma metodologia estatística que possibilitou obter estimativas de parâmetros ótimos para os algoritmos, além da efetiva comparação de performance das técnicas propostas. Adicionalmente, ferramentas de comparação multi-objetivo de conjuntos Pareto de aproximação foram elaboradas para a análise estatística adicional, assim como discussões sobre a generalização da análise estatística para especificações e configurações arbitrárias no projeto de filtros digitais.

As estimativas de parâmetros ótimos para o algoritmo genético e a otimização por enxame de partículas aplicados no projeto de um filtro rejeita-faixa com atraso de grupo reduzido nas bandas de passagem foram obtidas seguindo a análise exploratória estatística, assim como a comparação de performance entre os vários algoritmos de otimização. Para o problema particular de otimização proposto, o algoritmo com melhor performance (maior valor do indicador  $\epsilon$ -binário) foi o algoritmo genético, seguido do resfriamento simulado, otimização por enxame de partículas e por último, o algoritmo quasi-Newton.

Adicionalmente, os resultados para uma realização arbitrária dos algoritmos propostos no projeto do filtro FIR rejeita-faixa indicaram o algoritmo OEA como o que obteve os dois filtros com a melhor aproximação isolada da magnitude da resposta em frequência e do atraso de grupo do filtro. Esses resultados são confirmados em [29], onde os algoritmos de otimização propostos foram empregados na otimização de um filtro rejeita-faixa com as mesmas especificações do presente trabalho e foi reportado a otimização por enxame de partículas como a que obteve a melhor aproximação. Entretanto, em [29], a análise estatística comparativa foi feita considerando apenas o subproblema mono-objetivo da função objetivo em (2.47), particularmente para o conjunto de pesos com valor  $w_1 = \delta = 1$  e  $w_2 = (1 - \delta) = 0$ , ou seja, foi considerado apenas o problema de otimização mono-objetivo de minimização da distância de aproximação da magnitude da resposta em frequência do filtro.

Embora no presente trabalho o algoritmo otimização por enxame de partículas tenha obtido um conjunto Pareto ótimo com mais ampla cobertura de soluções diversas, algumas soluções intermediárias em particular do conjunto de aproximação possuíram valor bastante elevado da distância de aproximação, contribuindo para a deterioração das estatísticas do indicador  $\epsilon$ -binário no experimento comparativo. Notou-se ainda, particularmente com relação ao algoritmo quasi-Newton, que embora em algumas realizações tenha-se obtido valores bastante promissores do in-

dicador  $\epsilon$ -binário, em outras obteve-se péssimos valores. Pode-se inferir que esse comportamento foi observado devido ao aprisionamento local do algoritmo, quando o mesmo foi inicializado com um ponto inicial longe do ótimo global da função objetivo. Consequentemente, nota-se pelos resultados obtidos, a robustez que os algoritmos *soft-computing* possuem em se livrar de mínimos locais da função objetivo, mesmo quando inicializados com pontos bem distantes do ótimo global. Essa afirmativa é comprovada a partir das estatísticas superiores do indicador  $\epsilon$ -binário obtidas com os algoritmos *soft-computing* no experimento estatístico comparativo.

Finalmente, considerando a análise estatística com extensões multi-objetivo propostas no presente trabalho, o algoritmo genético foi a abordagem que levou a obtenção de conjuntos Pareto com as melhores estatísticas do indicador  $\epsilon$ -binário para o projeto do filtro FIR rejeita-faixa com atraso reduzido proposto. Por sua vez, o algoritmo OEA foi o algoritmo que obteve conjuntos Pareto-ótimo com melhor diversidade (ampla cobertura) de soluções diversas. Por fim, vale lembrar que os resultados comparativos obtidos no presente trabalho são mantidos apenas para a especificação do filtro FIR rejeita-faixa particular de teste proposto. Tendo em vista os resultados do teorema NFL e suas extensões obtidas na literatura, os resultados do presente trabalho são motivadores para uma possível generalização da aplicação dos algoritmos aos possíveis problemas de otimização multi-objetivo no projeto de filtros digitais, para assim, inferir um possível algoritmo de otimização que seja o melhor possível para esse domínio de aplicação. Entretanto é necessário teoria adicional para essa importante generalização, e esse seria um futuro campo de investigação adicional.

# REFERÊNCIAS BIBLIOGRÁFICAS

- [1] OPPENHEIM, A.; SCHAFER, R. *Discrete-Time Signal Processing, Second Edition*. [S.l.]: Prentice-Hall, Inc, 1999.
- [2] PROAKIS, J. G.; MANOLAKIS, D. G. *Digital Signal Processing, Principles, Algorithms and Applications, Third Edition*. [S.l.]: Prentice-Hall, Inc, 1996.
- [3] MADISETTI, V.; WILLIAMS, D. *Digital Signal Processing Handbook*. [S.l.]: CRC Press, 1999.
- [4] DINIZ, P.; SILVA, E. da; NETTO, S. *Processamento Digital de Sinais - Projeto e Análise de Sistemas*. [S.l.]: Bookman, 2004.
- [5] PARKS, T.; MCCLELLAN, J. Chebyshev approximation for nonrecursive digital filters with linear phase. *Circuits Theory, IEEE Transactions on [legacy, pre - 1988]*, v. 19, n. 2, p. 189–194, 1972. ISSN 0098-4094.
- [6] LIM, Y.-C. et al. A weighted least squares algorithm for quasi-equiripple fir and iir digital filter design. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, v. 40, n. 3, p. 551–558, 1992. ISSN 1053-587X.
- [7] ANTONIOU, A.; LU, W.-S. *Practical Optimization, Algorithms and Engineering Applications*. [S.l.]: Springer Science+Business Media, LLC, 2007.
- [8] DECZKY, A. Synthesis of recursive digital filters using the minimum p-error criterion. *Audio and Electroacoustics, IEEE Transactions on*, v. 20, n. 4, p. 257–263, 1972. ISSN 0018-9278.
- [9] DECZKY, A. Equiripple and minimax (chebyshev) approximations for recursive digital filters. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing]*, *IEEE Transactions on*, v. 22, n. 2, p. 98–111, 1974. ISSN 0096-3518.
- [10] FLETCHER, R. *Practical Methods of Optimization - Volume 1 - Unconstrained Optimization*. [S.l.]: John Wiley and Sons, 1980.
- [11] BAZARAA, M. S.; SHERALI, H. D.; SHETTY, C. M. *Nonlinear Programming - Theory and Algorithms, Third Edition*. [S.l.]: Wiley-Interscience, 2006.
- [12] LUENBERGER, D. *Linear and Nonlinear Programming*. [S.l.]: Addison-Wesley, 1984.
- [13] APOSTOL, T. *Calculus: One-Variable Calculus, with an Introduction to Linear Algebra - Second Edition*. [S.l.]: John Wiley & Sons, 1967.
- [14] NOCEDAL, J.; WRIGHT, S. *Numerical Optimization*. [S.l.]: Springer, 2006.



- [15] BORWEIN, J.; LEWIS, A. *Convex Analysis and Nonlinear Optimization - Theory and Examples*. [S.l.]: Springer-Verlag New York, Inc., 2000.
- [16] ENGELBRECHT, A. *Computational Intelligence An Introduction, Second Edition*. [S.l.]: John Wiley & Sons, Ltd, 2007.
- [17] BONNANS, J. et al. *Numerical Optimization – Theoretical and Practical Aspects, Second Edition*. [S.l.]: Springer Verlag, Berlin, 2006. (Universitext).
- [18] FLETCHER, R. *Practical Methods of Optimization - Volume 2 - Constrained Optimization*. [S.l.]: John Wiley and Sons, 1981.
- [19] MIETTINEN, K. *Nonlinear Multiobjective Optimization*. [S.l.]: Kluwer Academic Publishers, Dordrecht, 1999. (International Series in Operations Research and Management Science, v. 12).
- [20] SAWARAGI, Y.; NAKAYAMA, H.; TANINO, T. *Theory of Multiobjective Optimization*. [S.l.]: Academic Press, Orlando, 1985.
- [21] EHRGOTT, M.; GANDIBLEAUX, X. (Ed.). *Multiple Criteria Optimization. State of the Art Annotated Bibliographic Surveys*. [S.l.]: Kluwer Academic Publishers, Boston, 2002. (International Series in Operations Research and Management Science, v. 52).
- [22] CHANKONG, V.; HAIMES, Y. *Multiobjective Decision Making Theory and Methodology*. [S.l.]: Elsevier Science, New York, 1983.
- [23] LI, D. Convexification of a noninferior frontier. *J. Optim. Theory Appl.*, Plenum Press, New York, NY, USA, v. 88, n. 1, p. 177–196, 1996. ISSN 0022-3239.
- [24] REMEZ, E. I. General computational methods of chebyshev approximation: The problems with linear real parameters. *U.S. Atomic Energy Commission, Division of Technical Information*, 1962.
- [25] CORTELAZZO, G.; LIGHTNER, M. Simultaneous design in both magnitude and group-delay of iir and fir filters based on multiple criterion optimization. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing]*, *IEEE Transactions on*, v. 32, n. 5, p. 949–967, Oct 1984. ISSN 0096-3518.
- [26] CALVAGNO, G.; CORTELAZZO, G.; MIAN, G. A technique for multiple criterion approximation of fir filters in magnitude and group delay. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, v. 43, n. 2, p. 393–400, Feb 1995. ISSN 1053-587X.
- [27] AHMAD, S. U.; ANTONIOU, A. A multiobjective genetic algorithm for asymmetric fir filters. *Signal Processing and Information Technology, 2007 IEEE International Symposium on*, p. 525–530, 15–18 Dec. 2007.

- [28] TEIXEIRA, F.; ROMARIZ, A. Optimum finite impulse response digital filter design using computational intelligence based optimization algorithms. In: *ISDA '07: Proceedings of the Seventh International Conference on Intelligent Systems Design and Applications*. Washington, DC, USA: IEEE Computer Society, 2007. p. 635–640. ISBN 0-7695-2976-3.
- [29] TEIXEIRA, F.; ROMARIZ, A. Digital filter arbitrary magnitude and phase approximations - statistical analysis applied to a stochastic-based optimization approach. In: *to be published on Proceedings of the 2008 IEEE World Congress on Computational Intelligence*. [S.l.]: IEEE Press, 2008.
- [30] MORÉ, J. J.; THUENTE, D. J. Line search algorithms with guaranteed sufficient decrease. *ACM Trans. Math. Softw.*, ACM, New York, NY, USA, v. 20, n. 3, p. 286–307, 1994. ISSN 0098-3500.
- [31] DENNIS, J.; SCHNABEL, R. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. [S.l.]: Prentice Hall, 1983.
- [32] HE, J.; YAO, X. A study of drift analysis for estimating computation time of evolutionary algorithms. *Natural Computing: an international journal*, Kluwer Academic Publishers, Hingham, MA, USA, v. 3, n. 1, p. 21–35, 2004. ISSN 1567-7818.
- [33] KONAR, A. *Computational Intelligence - Principles, Techniques and Applications*. [S.l.]: Springer-Verlag, 2005.
- [34] PEDRYCZ, W. *Fuzzy Sets Engineering*. [S.l.]: CRC Press, 1996.
- [35] CZARN, A. et al. Statistical exploratory analysis of genetic algorithms. *Evolutionary Computation, IEEE Transactions on*, v. 8, n. 4, p. 405–421, Aug. 2004. ISSN 1089-778X.
- [36] RAHNAMAYAN, S.; TIZHOOSH, H. R.; SALAMA, M. M. A. A novel population initialization method for accelerating evolutionary algorithms. *Comput. Math. Appl.*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 53, n. 10, p. 1605–1614, 2007. ISSN 0898-1221.
- [37] MELANIE, M. *An Introduction to Genetic Algorithms*. [S.l.]: MIT Press, 1999.
- [38] MENG, Q. et al. Genetic algorithms encoding study and a sufficient convergence condition of gas. *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, v. 1, p. 649–652 vol.1, 1999.
- [39] OYAMA, A.; OBAYASHI, S.; NAKAMURA, T. Real-coded adaptive range genetic algorithm applied to transonic wing optimization. In: Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo, H.-P. S. (Ed.). *Parallel Problem Solving from Nature - PPSN VI 6th International Conference*. Paris, France: Springer Verlag, 2000.

- [40] CHAMBERS, L. (Ed.). *Practical Handbook of Genetic Algorithms, New Frontiers, vol. II*. [S.l.]: CRC Press, 1995.
- [41] PHAM, D.; JIN, G. Genetic algorithm using gradient-like reproduction operator. *Electronics Letters*, v. 31, n. 18, p. 1558–1559, 1995. ISSN 0013-5194.
- [42] DEB, K.; AGRAWAL, R. B. Simulated binary crossover for continuous search space. *Complex Systems*, v. 9, p. 115–148, 1995.
- [43] MICHALEWICZ, Z. *Genetic Algorithms Plus Data Structures Equals Evolution Programs*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1996. ISBN 0387580905.
- [44] HAFTKA, R.; GURDAL, Z. *Elements of Structural Optimization*. [S.l.]: Kluwer Academic Publishers, 1992.
- [45] NAM, D.; LEE, J.-S.; PARK, C. H. n-dimensional cauchy neighbor generation for the fast simulated annealing(algorithm theory). *IEICE transactions on information and systems*, The Institute of Electronics, Information and Communication Engineers, v. 87, n. 11, p. 2499–2502, 2004. ISSN 09168532.
- [46] MENDONCA, P.; CALOBA, L. New simulated annealing algorithms. *Circuits and Systems, 1997. ISCAS '97., Proceedings of 1997 IEEE International Symposium on*, v. 3, p. 1668–1671 vol.3, 1997.
- [47] METROPOLIS, N. et al. Equation of state calculations by fast computing machines. *jcp*, v. 21, p. 1087–1092, jun 1953.
- [48] INGBER, L. Very fast simulated re-annealing. *Mathematical Computer Modelling*, v. 12, n. 8, p. 967–973, 1989.
- [49] KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. *Science, Number 4598, 13 May 1983*, v. 220, 4598, p. 671–680, 1983.
- [50] KENNEDY, J.; EBERHART, R.; SHI, Y. *Swarm Intelligence*. [S.l.]: Morgan Kaufmann Publishers, 2001.
- [51] KENNEDY, J.; EBERHART, R. Particle swarm optimization. *Neural Networks, 1995. Proceedings., IEEE International Conference on*, v. 4, p. 1942–1948 vol.4, 1995.
- [52] JIAN, W.; XUE, Y.-C.; QIAN, J.-X. An improved particle swarm optimization algorithm with neighborhoods topologies. *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, v. 4, p. 2332–2337 vol.4, 2004.
- [53] EL-GALLAD, A. et al. Enhancing the particle swarm optimizer via proper parameters selection. *Electrical and Computer Engineering, 2002. IEEE CCECE 2002. Canadian Conference on*, v. 2, p. 792–797 vol.2, 2002. ISSN 0840-7789.

- [54] JIANG, M.; LUO, Y. P.; YANG, S. Y. Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. *Inf. Process. Lett.*, Elsevier North-Holland, Inc., Amsterdam, The Netherlands, The Netherlands, v. 102, n. 1, p. 8–16, 2007. ISSN 0020-0190.
- [55] PEER, E.; BERGH, F. van den; ENGELBRECHT, A. Using neighbourhoods with the guaranteed convergence pso. *Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE*, p. 235–242, 2003.
- [56] KENNEDY, J.; MENDES, R. Population structure and particle swarm performance. In: *CEC '02: Proceedings of the Evolutionary Computation on 2002. CEC '02. Proceedings of the 2002 Congress*. Washington, DC, USA: IEEE Computer Society, 2002. p. 1671–1676. ISBN 0-7803-7282-4.
- [57] ZITZLER, E. et al. Performance assessment of multiobjective optimizers: an analysis and review. *Evolutionary Computation, IEEE Transactions on*, v. 7, n. 2, p. 117–132, April 2003.
- [58] MONTGOMERY, D. *Design and Analysis of Experiments, 5th Edition*. [S.l.]: New York: Wiley, 1976.
- [59] COHEN, J. *Statistical Power Analysis for the Behavioral Sciences, Second Edition*. [S.l.]: Lawrence Erlbaum Associates, Publishers, 1988.
- [60] TIWARI, S. *Analysis of Unary and Binary Performance Indicators*. [S.l.], 2008. Disponível em: <<http://www.iitk.ac.in/kangal/seminar.shtml>>.
- [61] WOLPERT, D. H.; MACREADY, W. G. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, v. 1, n. 1, p. 67–82, April 1997.
- [62] IGEL, C.; TOUSSAINT, M. A no-free-lunch theorem for non-uniform distributions of target functions. *Journal of Mathematical Modelling and Algorithms*, Kluwer Academic Publishers, v. 3, n. 4, p. 313–322, January 2005. ISSN 1570-1166.
- [63] MACNISH, C. Towards unbiased benchmarking of evolutionary and hybrid algorithms for real-valued optimisation. *Connect. Sci.*, v. 19, n. 4, p. 361–385, 2007.
- [64] Sun Microsystems, Inc. *Java versão 1.6.0.05*. [S.l.], 2008. Disponível em: <<http://www.java.sun.com>>.
- [65] GAMMA, E. et al. *Design Patterns, Elements of Reusable Object-Oriented Software*. [S.l.]: Addison-Wesley Professional Computing Series, 1998.
- [66] BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *The Unified Modeling Language User Guide*. [S.l.]: Addison-Wesley Longman, Inc., 1999.
- [67] World Wide Web Consortium (W3C). *W3C XML Protocol Working Group*. [S.l.], 2008. Disponível em: <<http://www.w3.org>>.

- [68] MATHWORKS. *The Matlab distributed computing engine*. [S.l.], 2008. Disponível em: <<http://www.mathworks.com>>.
- [69] Insightful Corporation. *S-PLUS*. [S.l.], 2008. Disponível em: <<http://www.insightful.com>>.
- [70] PARRILO, P. A.; STURMFELS, B. Minimizing polynomial functions. *DIMACS SERIES IN DISCRETE MATHEMATICS AND THEORETICAL COMPUTER S*, 2003.
- [71] SEILER, P.; PARRILO, P. A. *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*. [S.l.], 2004. Disponível em: <<http://www.cds.caltech.edu/sostools>>.





# I. ALGORITMOS

## I.1 ALGORITMOS DE OTIMIZAÇÃO NÃO-LINEAR BASEADOS NO GRADIENTE

### I.1.1 Método Quasi-Newton

Essa subseção contém a descrição completa do algoritmo quasi-Newton discutido na seção 4.7 do Capítulo 4. O laço principal da rotina quasi-newton está definido no algoritmo I.1.1. Adicionalmente, as sub-rotinas de verificação do critério de parada do algoritmo de otimização, inicialização da matriz Hessiana na forma não-fatorada, cálculo do menor valor de representação da máquina e da atualização da estimativa da matriz Hessiana, se encontram respectivamente definidos nos algoritmos I.1.2, I.1.3, I.1.4 e I.1.5. As rotinas desenvolvidas são baseadas no *framework* de otimização não-linear baseado no gradiente desenvolvido em [31].

#### I.1.1.1 Funções auxiliares para algoritmo Quasi-Newton

Essa subseção contém as rotinas auxiliares chamadas a partir do algoritmo I.1.1.

### I.1.2 Algoritmo de Busca em Linha

Essa seção contém a descrição completa do algoritmo I.1.6 com a rotina principal de busca em linha para otimização não-linear unidimensional discutida na seção 4.3 do Capítulo 4. Esse algoritmo é baseado no *framework* de otimização não-linear delineado em [31]. As seguintes observações devem ser feitas com relação a esse algoritmo:

- No algoritmo I.1.6, linha I.1.6.3, a matriz  $D_x$  é a matriz diagonal de  $S_x$ . Por sua vez,  $S_x[i] = \frac{1}{typx[i]}$ , onde  $typx$  é um vetor  $n$  dimensional no qual o  $i$ -ésimo componente é um escalar positivo especificando a magnitude típica de  $x[i]$ , os coeficientes do filtro, conforme definidos na equação (2.37).
- A variável *passotol* é um escalar positivo representando a tolerância na qual a distância entre duas iterações sucessivas são consideradas perto o suficiente de zero. Na implementação do algoritmo, foi utilizado a sugestão [31]  $passotol = \epsilon_{maquina}^{2/3}$ , onde  $\epsilon_{maquina}$  é definido como o menor número positivo  $\tau$  tal que  $1 + \tau > 1$  representável na máquina de implementação do algoritmo.

Adicionalmente as sub-rotinas de redução do tamanho de passo  $\alpha$  com a criação de modelos quadráticos e cúbicos da função objetivo, e a geração de um tamanho de passo  $\alpha$  que atenda



---

**Algoritmo I.1.1: MinimizadorQuasiNewton()**

---

**Entrada:**  $n \in \mathbb{Z}$ ,  $\mathbf{x}_0 \in \mathbb{R}^n$ ,  $\nabla f$ ,  $\text{TypX} \in \mathbb{R}^n$ ,  $\text{TypF} \in \mathbb{R}$ ,  $f_{\text{digitos}} \in \mathbb{Z}$ ,  $\text{TolGrad} \in \mathbb{R}$ ,  
 $\text{TolPasso} \in \mathbb{R}$ ,  $\text{PassoMax} \in \mathbb{R}$ ,  $\text{LimIter} \in \mathbb{Z}$

**Saída:**  $\mathbf{x}_f \in \mathbb{R}^n$ ,  $\text{CodTerm} \in \mathbb{Z}$

```
I.1.1.1 CalcularEpsilonDaMaquina(EpsMq), algoritmo I.1.4;
I.1.1.2 ContIter  $\leftarrow$  0;
I.1.1.3 CalcularFuncaoDeCusto( $n, \mathbf{x}_0, f_c$ );
I.1.1.4 CalcularGradienteDaFuncaoDeCusto( $n, \mathbf{x}_0, \mathbf{g}_c$ );
I.1.1.5 VerificarCritérioDeParada( $n, \mathbf{x}_0, f_c, \mathbf{g}_c, \mathbf{S}_x, \text{TypF}, \text{TolGrad}, \text{MaxConsec}, \text{CodTerm}$ ),
    algoritmo I.1.2;
I.1.1.6 se CodTerm > 0 então
I.1.1.7    $\mathbf{x}_f \leftarrow \mathbf{x}_0$ ;
I.1.1.8 senão
I.1.1.9   InicializarHessianaNaoFatorada( $n, f_c, \text{TypF}, \mathbf{S}_x, \mathbf{B}_c$ ), algoritmo I.1.3;
I.1.1.10  $\mathbf{x}_c \leftarrow \mathbf{x}_0$ ;
I.1.1.11 enquanto CodTerm = 0 faça
I.1.1.12   ContIter  $\leftarrow$  ContIter + 1;
I.1.1.13    $\mathbf{p}_c = \mathbf{B}_c^{-1} \mathbf{g}_c$ ;
I.1.1.14   LineSearch( $n, \mathbf{x}_c, f_c, \mathbf{g}_c, \mathbf{p}_c, \mathbf{S}_x, \text{PassoMax}, \text{TolPasso}, \text{CodTerm}, \mathbf{x}_+, \mathbf{g}_+, \text{PassoMaxAlcancado}$ );
I.1.1.15   CalcularGradienteDaFuncaoDeCusto( $n, \mathbf{x}_+, \mathbf{g}_+$ );
I.1.1.16   VerificarCritérioDeParada( $n, \mathbf{x}_0, f_c, \mathbf{g}_c, \mathbf{S}_x, \text{TypF}, \text{TolGrad}, \text{MaxConsec}, \text{CodTerm}$ );
I.1.1.17   se CodTerm > 0 então
I.1.1.18      $\mathbf{x}_f \leftarrow \mathbf{x}_+$ ;
I.1.1.19   senão
I.1.1.20     AtualizarHessianaComBFGS( $n, \mathbf{x}_c, \mathbf{x}_+, \mathbf{g}_c, \mathbf{g}_+, \text{EpsMq}, \mathbf{B}_c$ ), algoritmo I.1.5;
I.1.1.21      $\mathbf{x}_c \leftarrow \mathbf{x}_+$ ;
I.1.1.22      $f_c \leftarrow f_+$ ;
I.1.1.23      $\mathbf{g}_c \leftarrow \mathbf{g}_+$ ;
```

---

---

**Algoritmo I.1.2:** VerificarCriterioDeParada( $n, \mathbf{x}_0, f_c, \mathbf{g}_c, \mathbf{S}_x, \text{TypF}, \text{GradTol}, \text{ConsecMax}, \text{CodTerm}$ )

---

**Entrada:**  $n \in \mathbb{Z}, \mathbf{x}_0 \in \mathbb{R}^n, f \in \mathbb{R}, \mathbf{g} \in \mathbb{R}^n, \mathbf{S}_x, \text{typf} \in \mathbb{R}, \text{TolGrad} \in \mathbb{R}$

**Saída:**  $\text{CodTerm} \in \mathbb{Z}, \text{PassoMaxConsecAlcancado} \in \mathbb{Z}$

```

I.1.2.1 CodTerm  $\leftarrow$  0;
I.1.2.2 se  $\max_{1 \leq i \leq n} \left\{ |g[i]| * \frac{\max\left\{ |x_+[i]|, \frac{1}{S_x[i]} \right\}}{\max\{|f|, \text{TypF}\}} \right\} \leq \text{TolGrad}$  então
I.1.2.3   | CodTerm  $\leftarrow$  1;
I.1.2.4   senão se  $\max_{1 \leq i \leq n} \left\{ \frac{|x_+[i] - x_c[i]|}{\max\left\{ |x_+[i]|, \frac{1}{S_x[i]} \right\}} \right\}$  então
I.1.2.5     | CodTerm  $\leftarrow$  2;
I.1.2.6   senão se  $\text{Contlter} \geq \text{Limlter}$  então
I.1.2.7     | CodTerm  $\leftarrow$  4;
I.1.2.8   senão se  $\text{PassoMaxAlcancado} = \text{VERDADEIRO}$  então
I.1.2.9     | PassoMaxConsecAlcancado  $\leftarrow$  PassoMaxConsecAlcancado + 1;
I.1.2.10    se  $\text{PassoMaxConsecAlcancado} = 5$  então
I.1.2.11    | CodTerm  $\leftarrow$  5;
I.1.2.12   senão
I.1.2.13    | PassoMaxConsecAlcancado  $\leftarrow$  0;

```

---



---

**Algoritmo I.1.3:** InicializarHessianaNaoFatorada( $n, f_c, \text{TypF}, \mathbf{S}_x$ )

---

**Entrada:**  $n \in \mathbb{Z}, f \in \mathbb{R}, \text{typf} \in \mathbb{R}, \mathbf{S}_x \in \mathbb{R}$

**Saída:**  $\mathbf{B} \in \mathbb{R}^{n \times n}$  simétrica

```

I.1.3.1 Temp  $\leftarrow$   $\max\{|f|, \text{TypF}\}$ ;
I.1.3.2 para  $i = 1$  até  $n$  faça
I.1.3.3   |  $B[i, i] \leftarrow \text{Temp} * S_x[i] * S_x[i]$ ;
I.1.3.4   para  $j = i + 1$  até  $n$  faça
I.1.3.5     |  $H[i, j] \leftarrow 0$ ;

```

---



---

**Algoritmo I.1.4:** CalcularEpsilonDaMaquina()

---

**Entrada:** nenhuma

**Saída:**  $\text{EpsMaq} \in \mathbb{R}$

```

I.1.4.1 EpsMaq  $\leftarrow$  1;
I.1.4.2 repita
I.1.4.3   |  $\text{EpsMaq} \leftarrow \frac{\text{EpsMaq}}{2}$ 
I.1.4.4 até  $(1 + \text{EpsMaq}) = 1$ ;
I.1.4.5 EpsMaq  $\leftarrow 2 * \text{EpsMaq}$ 

```

---

---

**Algoritmo I.1.5:** AtualizarHessianaComBFGS(  $n, \mathbf{x}_c, \mathbf{x}_+, \mathbf{g}_c, \mathbf{g}_+, \text{EpsMq}, B_c$  )

---

**Entrada:**  $n \in \mathbb{Z}, \mathbf{x}_c \in \mathbb{R}^n, \mathbf{x}_+ \in \mathbb{R}^n, \mathbf{g}_c \in \mathbb{R}^n, \mathbf{g}_+ \in \mathbb{R}^n, \text{EpsMq} \in \mathbb{R}$

**Saída:**  $B \in \mathbb{R}^{n \times n}$

```
I.1.5.1  $\mathbf{s} \leftarrow \mathbf{x}_+ - \mathbf{x}_c;$ 
I.1.5.2  $\mathbf{y} \leftarrow \mathbf{g}_+ - \mathbf{g}_c;$ 
I.1.5.3  $\text{temp1} \leftarrow \mathbf{y}^T \mathbf{s};$ 
I.1.5.4 se  $\text{temp1} \geq (\text{EpsMq})^{\frac{1}{2}} \|\mathbf{s}\|_2 \|\mathbf{y}\|_2$  então
I.1.5.5      $\text{Tol} \leftarrow \eta;$ 
I.1.5.6      $\text{PularAtualizacao} \leftarrow \text{VERDADEIRO};$ 
I.1.5.7     para  $i = 1$  até  $n$  faça
I.1.5.8          $t[i] \leftarrow \sum_{j=1}^i B[j, i] * s[j] + \sum_{j=i+1}^n B[i, j] * s[j];$ 
I.1.5.9         se  $|y[i] - t[i]| \geq \text{Tol} * \max\{|g_c[i]|, |g_+[i]|\}$  então
I.1.5.10              $\text{PularAtualizacao} \leftarrow \text{FALSO};$ 
I.1.5.11 se  $\text{PularAtualizacao} = \text{FALSO}$  então
I.1.5.12      $\text{temp2} \leftarrow \mathbf{s}^T \mathbf{t};$ 
I.1.5.13     para  $i = 1$  até  $n$  faça
I.1.5.14         para  $j = 1$  até  $n$  faça
I.1.5.15              $B[i, j] \leftarrow B[i, j] + \frac{y[i]*y[j]}{\text{temp1}} - \frac{t[i]*t[j]}{\text{temp2}};$ 
```

---

a condição de curvatura, se encontram definidos respectivamente nos algoritmos I.1.7 e I.1.8.

### I.1.2.1 Funções Auxiliares para o Algoritmo de Busca em Linha

Essa seção contém as rotinas auxiliares chamadas a partir do algoritmo I.1.6.

## I.2 ALGORITMOS DE OTIMIZAÇÃO SOFT-COMPUTING

Essa seção contém a descrição completa dos algoritmos de otimização *soft-computing* apresentados no Capítulo 5. Mais especificamente, a subseção I.2.1 apresenta a descrição do algoritmo genético. A subseção I.2.2 apresenta descrição do algoritmo resfriamento simulado e por fim, a subseção I.2.3 apresenta a descrição do algoritmo otimização por enxame de partículas.

### I.2.1 Algoritmo Genético

Essa subseção contém a descrição do algoritmo genético definido na seção 5.4 do Capítulo 5. Convém notar com relação ao algoritmo genético definido no algoritmo I.2.1 que o laço principal

---

**Algoritmo I.1.6:** LineSearch(  $n, \mathbf{x}_c, f_c, \mathbf{g}_c, \mathbf{p}_c, \mathbf{S}_x$ , PassoMax, TolPasso, CodTerm,  $\mathbf{x}_+$ ,  $\mathbf{g}_+$ , PassoMaxAlcancado )

---

**Entrada:**  $n \in \mathbb{Z}, \mathbf{x}_c \in \mathbb{R}^n, f_c \in \mathbb{R}, \nabla f, \mathbf{p} \in \mathbb{R}^n, \mathbf{S}_x \in \mathbb{R}^n$ , PassoMax  $\in \mathbb{R}$ , TolPasso  $\in \mathbb{R}$

**Saída:**  $\mathbf{x}_+ \in \mathbb{R}^n, f_+ \in \mathbb{R}$ , PassoMaxAlcancado  $\in$  Booleano

I.1.6.1 PassoMaxAlcancado  $\leftarrow$  FALSO;

I.1.6.2  $\mu \leftarrow 10^{-4}$ ;

I.1.6.3 PassoNewton  $\leftarrow \|D_x p\|_2$ ;

I.1.6.4 se PassoNewton  $>$  PassoMax então

I.1.6.5  $p \leftarrow p \frac{\text{PassoMax}}{\text{PassoNewton}}$ ;

I.1.6.6  $\text{PassoNewton} \leftarrow \text{PassoMax}$ ;

I.1.6.7 CurvaturaInicial  $\leftarrow g^T p$ ;

I.1.6.8  $\text{TamRelDirecao} \leftarrow \max_{1 \leq i \leq n} \left[ \frac{|p[i]|}{\max(|x_0[i]|, \frac{1}{S_x[i]})} \right]$ ;

I.1.6.9  $\alpha_{min} \leftarrow \frac{\text{TolPasso}}{\text{TamRelDirecao}}$ ;

I.1.6.10  $\alpha \leftarrow 1$  ;

I.1.6.11 repita

I.1.6.12  $x_+ \leftarrow x_0 + \alpha p$ ;

I.1.6.13  $f_+ \leftarrow \text{CalcularFuncaoDeCusto}(x_+)$ ;

I.1.6.14 se  $f_+ \leq f_0 + \mu \alpha * \text{CurvaturaInicial}$  então

I.1.6.15  $g_+ \leftarrow \text{CalcularGradienteDaFuncaoDeCusto}(x_+)$ ;

I.1.6.16  $\eta = 0.9$ ;

I.1.6.17  $\text{NovaCurvatura} \leftarrow g_+^T p$ ;

I.1.6.18 GerarAlphaComCondicaoDeCurvatura(), algoritmo I.1.8;

I.1.6.19 se  $\alpha * \text{PassoNewton} > 0.99 * \text{PassoMax}$  então

I.1.6.20  $\text{PassoMaxAlcancado} \leftarrow \text{VERDADEIRO}$ ;

I.1.6.21  $\text{return } \alpha$  ;

I.1.6.22 senão se  $\alpha < \alpha_{min}$  então

I.1.6.23  $\text{return } \alpha_{min}$ ;

I.1.6.24 senão

I.1.6.25  $\text{ReduzirAlpha}()$ ;

I.1.6.26 até Valor de  $\alpha$  seja retornado ;

---

---

**Algoritmo I.1.7:** ReduzirAlpha()**Entrada:** Mesmas que o algoritmo I.1.6**Saída:**  $\alpha \in \mathbb{R}$ **I.1.7.1 se**  $\alpha = 1$  **então****I.1.7.2**  $\alpha_{temp} \leftarrow -\frac{\text{CurvaturaInicial}}{2(f_+ - f_0 - \text{CurvaturaInicial})};$ **I.1.7.3 senão****I.1.7.4** 
$$\begin{bmatrix} a \\ b \end{bmatrix} \leftarrow \frac{1}{\alpha - \alpha_{ant}} \begin{bmatrix} \frac{1}{\alpha^2} & \frac{-1}{\alpha_{ant}^2} \\ \frac{-\alpha_{ant}}{\alpha^2} & \frac{\alpha}{\alpha_{ant}^2} \end{bmatrix} \begin{bmatrix} f_+ - f_0 - \alpha * \text{CurvaturaInicial} \\ f_{+ant} - f_0 - \alpha_{ant} * \text{CurvaturaInicial} \end{bmatrix};$$
**I.1.7.5**  $\text{Disco} \leftarrow b^2 - 3a * \text{CurvaturaInicial};$ **I.1.7.6 se**  $a = 0$  **então****I.1.7.7**  $\alpha_{temp} \leftarrow -\frac{\text{CurvaturaInicial}}{2b};$ **I.1.7.8 senão****I.1.7.9**  $\alpha_{temp} \leftarrow \frac{-b + \text{Disco}^{1/2}}{3a};$ **I.1.7.10 se**  $\alpha_{temp} > 0.5\alpha$  **então****I.1.7.11**  $\alpha_{temp} \leftarrow 0.5\alpha;$ **I.1.7.12**  $\alpha_{ant} \leftarrow \alpha;$ **I.1.7.13**  $f_{+ant} \leftarrow f_+;$ **I.1.7.14 se**  $\alpha_{temp} \leq 0.1\alpha$  **então****I.1.7.15**  $\alpha \leftarrow 0.1\alpha;$ **I.1.7.16 senão****I.1.7.17**  $\alpha \leftarrow \alpha_{temp};$ 

---

---

**Algoritmo I.1.8: GerarAlphaComCondicaoDeCurvatura()**

---

**Entrada:** Mesmas que o algoritmo I.1.6

**Saída:**  $\alpha \in \mathbb{R}$

```
I.1.8.1 se NovaCurvatura <  $\eta$  * CurvaturaInicial então
I.1.8.2   se  $\alpha = 1$  e PassoNewton < PassoMax então
I.1.8.3      $\alpha_{max} \leftarrow \frac{\text{PassoMax}}{\text{PassoNewton}}$ ;
I.1.8.4     repita
I.1.8.5        $\alpha_{ant} \leftarrow \alpha$ ;
I.1.8.6        $f_{+ant} \leftarrow f_+$ ;
I.1.8.7        $\alpha \leftarrow \min(2\alpha, \alpha_{max})$ ;
I.1.8.8        $x_+ \leftarrow x_0 + \alpha p$ ;
I.1.8.9        $f_+ \leftarrow \text{CalcularFuncaoDeCusto}(x_+)$ ;
I.1.8.10      se  $f_+ \leq f_0 + \mu\alpha$  * CurvaturaInicial então
I.1.8.11         $g_+ \leftarrow \text{CalcularGradienteDaFuncaoDeCusto}(x_+)$ ;
I.1.8.12        NovaCurvatura  $\leftarrow g_+^T p$ ;
I.1.8.13      até ( $f_+ > f_0 + \mu\alpha$  * CurvaturaInicial) ou (NovaCurvatura  $\geq \eta$  * CurvaturaInicial) ou ( $\alpha \geq \alpha_{max}$ );
I.1.8.14 se ( $\alpha < 1$ ) ou (( $\alpha > 1$ ) e ( $f_+ > f_0 + \mu\alpha$  * CurvaturaInicial)) então
I.1.8.15    $\alpha_{inf} \leftarrow \min(\alpha, \alpha_{ant})$ ;
I.1.8.16    $\alpha_{diff} \leftarrow |\alpha_{ant} - \alpha|$ ;
I.1.8.17   se  $\alpha < \alpha_{ant}$  então
I.1.8.18      $f_{inf} \leftarrow f_+$ ;
I.1.8.19      $f_{sup} \leftarrow f_{+ant}$ ;
I.1.8.20   senão
I.1.8.21      $f_{ant} \leftarrow f_{+prev}$ ;
I.1.8.22      $f_{sup} \leftarrow f_+$ ;
I.1.8.23   repita
I.1.8.24      $\alpha_{incr} \leftarrow \frac{-\text{NovaCurvatura} * (\alpha_{diff}^2)}{2(f_{sup} - (f_{inf} + \text{NovaCurvatura} * \alpha_{diff}))}$ ;
I.1.8.25     se  $\alpha_{incr} < 0.2\alpha_{diff}$  então
I.1.8.26        $\alpha_{incr} \leftarrow 0.2\alpha_{diff}$ ;
I.1.8.27      $\alpha \leftarrow \alpha_{inf} + \alpha_{incr}$ ;
I.1.8.28      $x_+ \leftarrow x_0 + \alpha p$ ;
I.1.8.29      $f_+ \leftarrow \text{CalcularFuncaoDeCusto}(x_+)$ ;
I.1.8.30     se  $f_+ > f_0 + \mu\alpha$  * CurvaturaInicial então
I.1.8.31        $\alpha_{diff} \leftarrow \alpha_{incr}$ ;
I.1.8.32        $f_{sup} \leftarrow f_+$ ;
I.1.8.33     senão
I.1.8.34        $g_+ \leftarrow \text{CalcularGradienteDaFuncaoDeCusto}(x_+)$ ;
I.1.8.35       NovaCurvatura  $\leftarrow g_+^T p$ ;
I.1.8.36       se NovaCurvatura <  $\eta$  * CurvaturaInicial então
I.1.8.37          $\alpha_{inf} \leftarrow \alpha$ ;
I.1.8.38          $\alpha_{diff} \leftarrow \alpha_{diff} - \alpha_{incr}$ ;
I.1.8.39          $f_{inf} \leftarrow f_+$ ;
I.1.8.40   até (NovaCurvatura  $\geq \eta$  * CurvaturaInicial) ou ( $\alpha_{diff} < \alpha_{min}$ );
I.1.8.41 se NovaCurvatura <  $\eta$  * CurvaturaInicial então
I.1.8.42    $\alpha \leftarrow \alpha_{inf}$ ;
```

---

é iterado enquanto o critério de parada não seja alcançado. Mais especificamente, esse critério é verificado através do algoritmo I.1.2, o mesmo utilizado na rotina de otimização quasi-Newton. Entretanto, como não é utilizada informação do gradiente no algoritmo genético, somente as verificações da diferença mínima entre iterações consecutivas, teste condicional I.1.2.4 do algoritmo I.1.2 e o número máximo de iterações, teste condicional I.1.2.6 do algoritmo I.1.2 são utilizados como critério de parada. Adicionalmente, no algoritmo I.2.1, a variável  $N$  é equivalente ao tamanho da população de indivíduos.

---

**Algoritmo I.2.1:** AlgoritmoGenetico()

---

**Entrada:**  $N \in \mathbb{Z}, p_{cr} \in \mathbb{R}, p_{mu} \in \mathbb{R}$

**Saída:**  $\mathbf{x}_f \in \mathbb{R}^n$

```

I.2.1.1 InicializarPopulacao(), algoritmo 5.2.1;
I.2.1.2 Calcular função de adequação  $f(\mathbf{X})$  de cada cromossomo na população;
I.2.1.3 repita
I.2.1.4   para  $i = 1$  até  $N$  faça
I.2.1.5     SelecaoRouletteWheel(), algoritmo 5.4.1 ;
I.2.1.6   para  $i = 1$  até  $\frac{N}{2}$  faça
I.2.1.7     Indivíduo1 = PopulacaoDeIndividuos( $2 * i - 1$ );
I.2.1.8     Indivíduo2 = PopulacaoDeIndividuos( $2 * i$ );
I.2.1.9     Com probabilidade  $p_{cr}$ , efetuar crossover em Indivíduo1 e Indivíduo2 de acordo com
           equação (5.8) ;
I.2.1.10    Com probabilidade  $p_{mu}$ , efetuar mutação de acordo com equação (5.10);
I.2.1.11    Substituir população corrente com a nova população ;
I.2.1.12 até Critério de parada seja alcançado ;

```

---

## I.2.2 Resfriamento Simulado

Essa subseção contém a descrição do algoritmo resfriamento simulado definido na seção 5.5 do Capítulo 5. Convém notar com relação ao algoritmo RS definido no algoritmo I.2.2 que o laço principal é iterado enquanto o critério de parada não seja alcançado. Mais especificamente, esse critério é verificado através do algoritmo I.1.2, o mesmo utilizado na rotina de otimização quasi-Newton. Entretanto, como não é utilizada informação do gradiente no algoritmo resfriamento simulado, somente as verificações da diferença mínima entre iterações consecutivas, teste condicional I.1.2.4 do algoritmo I.1.2 e o número máximo de iterações, teste condicional I.1.2.6 do algoritmo I.1.2 são utilizados como critério de parada. Adicionalmente, no algoritmo I.2.2, a variável com o contador de mudança de temperatura é equivalente ao índice temporal  $k$  da equação de resfriamento da temperatura do sistema em (5.14).

---

**Algoritmo I.2.2: SimulatedAnnealing()**

---

**Entrada:**  $\mathbf{x}_0 \in \mathbb{R}^n, T_0$

**Saída:**  $\mathbf{x}_f \in \mathbb{R}^n$

```
1.2.2.1 ContadorMudancaTemperatura  $\leftarrow$  1;
1.2.2.2 repita
1.2.2.3   repita
1.2.2.4     Perturbar ConfiguracaoAtualDosAtomos de acordo com equação (5.12);
1.2.2.5     Calcular probabilidade de aceitação do novo estado energético com equação (5.13);
1.2.2.6     se  $U(0, 1) \leq \text{probabilidade de aceitação}$  então
1.2.2.7       Equilibrio térmico atingido;
1.2.2.8   até equilibrio térmico seja atingindo ;
1.2.2.9   ContadorMudancaTemperatura  $\leftarrow$  ContadorMudancaTemperatura + 1;
1.2.2.10  Reduzir temperatura do sistema de acordo com a equação (5.14);
1.2.2.11 até Critério de parada seja alcançado ;
```

---

### I.2.3 Otimização por Enxame de Partículas

Essa subseção contém a descrição do algoritmo OEA definido na seção 5.6 do Capítulo 5. Convém notar com relação ao algoritmo OEA definido no algoritmo I.2.3 que o laço principal é iterado enquanto o critério de parada não seja alcançado. Mais especificamente, esse critério é verificado através do algoritmo I.1.2, o mesmo utilizado na rotina de otimização quasi-Newton. Entretanto, como não é utilizada informação do gradiente no algoritmo otimização por enxame de partículas, somente as verificações da diferença mínima entre iterações consecutivas, teste condicional I.1.2.4 do algoritmo I.1.2 e o número máximo de iterações, teste condicional I.1.2.6 do algoritmo I.1.2 são utilizados como critério de parada. Adicionalmente no algoritmo I.2.3, a variável  $N$  é equivalente ao tamanho da população de partículas e o laço com iteração do índice dos vizinhos da partícula corrente é definido com uma topologia em anel, conforme descrito na subseção 5.6.2 do Capítulo 5.



---

**Algoritmo I.2.3:** ParticleSwarmOptimization()

---

**Entrada:**  $n \in \mathbb{Z}, \phi_1 \in \mathbb{R}, \phi_2 \in \mathbb{R}, N \in \mathbb{Z}$

**Saída:**  $\mathbf{x}_f \in \mathbb{R}^n$

I.2.3.1 InicializarPopulacao(), algoritmo 5.2.1;

I.2.3.2 **repita**

I.2.3.3     **para**  $i = 1$  até  $N$  **faça**

I.2.3.4         **se**  $G(\vec{x}_i) > G(\vec{p}_i)$  **então**

I.2.3.5              $p_{id} = x_{id};$

I.2.3.6          $g = i;$

I.2.3.7         **para**  $j = \text{índices dos vizinhos}$  **faça**

I.2.3.8             **se**  $G(\vec{p}_j) > G(\vec{p}_g)$  **então**

I.2.3.9                  $g = j;$

I.2.3.10         **para**  $d = 1$  até  $n$  **faça**

I.2.3.11              $v_i(t) = v_{id}(t-1) + \phi_1(p_{id} - x_{id}(t-1)) + \phi_2(p_{gd} - x_{id}(t-1));$

I.2.3.12              $v_{id} \in (-V_{\max}, V_{\max});$

I.2.3.13              $x_{id}(t) = x_{id}(t-1) + v_{id}(t);$

I.2.3.14 **até** Critério de parada seja alcançado ;

---

## **II. FRAMEWORK DE OTIMIZAÇÃO**

### **II.1 EXEMPLOS DE ARQUIVOS DE CONFIGURAÇÃO DO FRAMEWORK DE OTIMIZAÇÃO**

---

## Trecho de Arquivo de Configuração 1 Exemplo de especificação de filtro rejeita-faixa com atraso de grupo reduzido

---

```
<Filtro tipo="FIRFaseNaoLinear">
<id>1</id>
<nome>Passa baixa com atraso de grupo reduzido</nome>
<descricao>
Especificação de filtro utilizada nos resultados da dissertação ENE.
</descricao>
<ordem>8</ordem>
<numeroMaximoAvaliacoesNaOtimizacao>20000</numeroMaximoAvaliacoesNaOtimizacao>
<modulo>
<normaLP>2</normaLP>
<Especificacao>
<FrequenciaInicial>0</FrequenciaInicial>
<FrequenciaFinal>0.25</FrequenciaFinal>
<Peso>1</Peso>
<Valor>0</Valor>
</Especificacao>
<Especificacao>
<FrequenciaInicial>0.3</FrequenciaInicial>
<FrequenciaFinal>0.65</FrequenciaFinal>
<Peso>1</Peso>
<Valor>-30</Valor>
</Especificacao>
<Especificacao>
<FrequenciaInicial>0.7</FrequenciaInicial>
<FrequenciaFinal>1</FrequenciaFinal>
<Peso>1</Peso>
<Valor>0</Valor>
</Especificacao>
</modulo>
<fase>
<normaLP>2</normaLP>
<Especificacao>
<FrequenciaInicial>0</FrequenciaInicial>
<FrequenciaFinal>0.25</FrequenciaFinal>
<Peso>1</Peso>
<Valor>0</Valor>
</Especificacao>
<Especificacao>
<FrequenciaInicial>0.7</FrequenciaInicial>
<FrequenciaFinal>1</FrequenciaFinal>
<Peso>1</Peso>
<Valor>0</Valor>
</Especificacao>
</fase>
</Filtro>
```

---

---

### Trecho de Arquivo de Configuração 2 Exemplo de configuração do algoritmo genético

---

```
<Algoritmo tipo="AlgoritmoPonderadoDeEscalarizacaoComAlgoritmoGenetico">
<id>1</id>
<nome>Algoritmo Ponderado de Escalarização com Algoritmo Genético</nome>
<descricao>GA com parâmetros ajustados para filtro rejeita-faixa com atraso de grupo reduzido</descricao>

<NumeroSolucoesDoConjuntoPareto>10</NumeroSolucoesDoConjuntoPareto>

<TamanhoPopulacao>20</TamanhoPopulacao>
<TaxaCrossover>0.7</TaxaCrossover>
<TaxaMutacao>0.15</TaxaMutacao>
<TaxaElitismo>0.1</TaxaElitismo>
<NumeroMaximoGeracoesEstagnadas>50</NumeroMaximoGeracoesEstagnadas>
<VarianciaOperadorMutacao>0.0001</VarianciaOperadorMutacao>
<VarianciaInicializacaoIndividuos>4</VarianciaInicializacaoIndividuos>
</Algoritmo>
```

---

---

### Trecho de Arquivo de Configuração 3 Exemplo de configuração do resfriamento simulado

---

```
<Algoritmo tipo="AlgoritmoPonderadoDeEscalarizacaoComSimulatedAnnealing">
<id>2</id>
<nome>Algoritmo Ponderado de Escalarização com o Resfriamento Simulado</nome>
<descricao>RS com parâmetros ajustados para filtro rejeita-faixa com atraso de grupo reduzido</descricao>

<NumeroSolucoesDoConjuntoPareto>10</NumeroSolucoesDoConjuntoPareto>

<VarianciaGeracaoEstadoInicial>4</VarianciaGeracaoEstadoInicial>
<TemperaturaInicial>0.0001</TemperaturaInicial>
<NumeroMaximoDeIteracoesEmEstadoEnergeticoEstagnado>10000</NumeroMaximoDeIteracoesEmEstadoEnergeticoEstagnado>
</Algoritmo>
```

---

---

### Trecho de Arquivo de Configuração 4 Exemplo de configuração da otimização por enxame de partículas

---

```
<Algoritmo tipo="AlgoritmoPonderadoDeEscalarizacaoComParticleSwarmOptimization">
<id>3</id>
<nome>Algoritmo Ponderado de Escalarização com a Otimização por Enxame de Partículas</nome>
<descricao>OEA com parâmetros ajustados para filtro rejeita-faixa com atraso de grupo reduzido</descricao>

<NumeroSolucoesDoConjuntoPareto>10</NumeroSolucoesDoConjuntoPareto>

<NumeroDeParticulas>20</NumeroDeParticulas>
<ConstantePhi>3.4</ConstantePhi>
<MomentoDeInercia>0.5</MomentoDeInercia>
<NumeroMaximoGeracoesEstagnadas>50</NumeroMaximoGeracoesEstagnadas>
<VarianciaInicializacaoParticulas>0.5</VarianciaInicializacaoParticulas>
</Algoritmo>
```

---

---

## Trecho de Arquivo de Configuração 5 Exemplo de configuração de experimento estatístico

---

```
<Experimento tipo="OtimizacaoParametrosAlgoritmoGenetico">
<id>1</id>
<nome>Diagrama de Pontos para Algoritmo Genético com Método Ponderado de Escalarização MultiObjetivo.</nome>
<descricao>Experimento para geração de diagrama de pontos do algoritmo genético para experimento na dissertação
</descricao>

<Algoritmo>
<NumeroSolucoesDoConjuntoPareto>10</NumeroSolucoesDoConjuntoPareto>

<TamanhoPopulacao>20</TamanhoPopulacao>
<TaxaElitismo>0.1</TaxaElitismo>
<NumeroMaximoGeracoesEstagnadas>50</NumeroMaximoGeracoesEstagnadas>
<VarianciaOperadorMutacao>0.0001</VarianciaOperadorMutacao>
<VarianciaInicializacaoIndividuos>4</VarianciaInicializacaoIndividuos>
</Algoritmo>

<TamanhoAmostral>1</TamanhoAmostral>
<Fator tipo="TaxaCrossover">
<Minimo>0</Minimo>
<Maximo>1</Maximo>
<Incremento>0.1</Incremento>
</Fator>
<Fator tipo="TaxaMutacao">
<Minimo>0</Minimo>
<Maximo>1</Maximo>
<Incremento>0.1</Incremento>
</Fator>
</Experimento>
```

---

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)