

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

TÍTULO DO TRABALHO

Síntese de grades de Bragg em fibra: técnicas de
aceleração e codificação para algoritmos evolucionários

NOME DO AUTOR

Marco José de Sousa

TD_01/2008

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2008

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

NOME DO AUTOR

Marco José de Sousa

TÍTULO DO TRABALHO

Síntese de grades de Bragg em fibra: técnicas de
aceleração e codificação para algoritmos evolucionários

TD_01/2008

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2008

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

NOME DO AUTOR

Marco José de Sousa

TÍTULO DO TRABALHO

**Síntese de grades de Bragg em fibra: técnicas de
aceleração e codificação para algoritmos evolucionários**

Tese submetida à Banca Examinadora
do Programa de Pós-graduação em En-
genharia Elétrica da UFPA para a
obtenção do Grau de Doutor em En-
genharia Elétrica.

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2008

SÍNTESE DE GRADES DE BRAGG EM FIBRA: TÉCNICAS DE ACELERAÇÃO E CODIFICAÇÃO PARA ALGORITMOS EVOLUCIONÁRIOS

C376e Sousa, Marco José de

Síntese de grades de Bragg em fibra: técnicas de aceleração e codificação para algoritmos evolucionários / Marco José de Sousa; orientador, João Crisóstomo Weyl Albuquerque Costa.-2008.

Tese (Doutorado) -Universidade Federal do Pará. Instituto de Tecnologia. Programa de Pós-Graduação em Engenharia Elétrica, Belém, 2008.

1. Fibras óticas. 2. Algoritmo genéticos. 3. Computação evolucionária. I. Título.

CDD - 22. ed. 621.3827

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

SÍNTESE DE GRADES DE BRAGG EM FIBRA: TÉCNICAS DE ACELERAÇÃO E CODIFICAÇÃO PARA ALGORITMOS EVOLUCIONÁRIOS

AUTOR: Marco José de Sousa

TESE DE DOUTORADO SUBMETIDA À AVALIAÇÃO DA BANCA EXAMINADORA APROVADA PELO COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA DA UNIVERSIDADE FEDERAL DO PARÁ E JULGADA ADEQUADA PARA OBTENÇÃO DO GRAU DE DOUTOR EM ENGENHARIA ELÉTRICA NA ÁREA DE TELECOMUNICAÇÃO.

APROVADA EM 11/02/2008

BANCA EXAMINADORA:

Prof. Dr. João Crisóstomo Weyl Albuquerque Costa (ORIENTADOR - UFPA)

Prof. Dr. José Luis Campos de Oliveira Santos (FCUP – INESC)

Prof. Dr. Licinius Dmitri de Alcântara (UFPA)

Prof. Dr. Marcelo Eduardo Vieira Segatto (UFES)

Prof. Dr. Maria José Pontes (UFES)

Prof. Dr. Maria Thereza Mirando Rocco Giraldi (IME)

Prof. Dr. Petrus Agripino de Alcântara (UFPA)

Prof. Dr. Roberto Célio Limão (UFPA)

Prof. Dr. Vladimiro Henrique Barrosa Pinto de Miranda (FEUP – INESC)

VISTO:

Prof. Dr. Evaldo Gonçalves Pelaes

COORDENADOR DO PPGE/ITEC/UFPA

Agradecimentos

Agradeço a Deus.

Agradeço à minha família.

Agradeço especialmente ao professor Dr. João Crisóstomo, pela imensa paciência desde o ano de 1999.

Aos professores Dr. José Luis dos Santos e Dr. Vladimiro Miranda, pela amizade e pela enriquecedora experiência oferecida no Instituto de Engenharia de Sistemas e Computadores do Porto (INESC-Porto) em Portugal.

Agradeço aos professores Dr. Aldebaro Klautau, Dr. Marcelo Vieira Segatto e Dr. Petrus Agrippino, pelo auxílio técnico e pela amizade.

Aos membros da minha banca examinadora de Tese, pela paciência, pelas contribuições técnicas e pela amizade.

Aos meus colegas André Cavalcante, Claudomiro Sales, Gilvan Borges, João Chamma, Josiane Rodrigues e Lamartine Souza, pela experiência profissional e acadêmica positiva no Laboratório de Eletromagnetismo Aplicado (LEA) e principalmente pela amizade.

Agradeço aos meus colegas e amigos desde 1997 do LEA e do Laboratório de Processamento de Sinais (LAPS).

Aos demais amigos e colegas do Laboratório de Engenharia Elétrica desde 1997.

Agradeço aos meus bons professores do curso de Engenharia Elétrica e do curso de Engenharia da Computação da Universidade Federal do Pará, especialmente os professores Eurípedes Pinheiro, Evaldo Pelaes, Gervásio Cavalcante, Renato Francês, Roberto Limão e Rubem Farias.

Resumo

As grades de Bragg em fibra (*Fiber Bragg Grating* – FBG) são componentes versáteis, sendo empregadas intensivamente como base dos dispositivos que constituem as redes ópticas. Algumas aplicações podem exigir dispositivos singulares dotados de grades de Bragg com características espectrais únicas. Dependendo do tipo de resposta espectral exigido de uma FBG, seu projeto pode ser extremamente difícil, exigindo técnicas computacionais especializadas. Devido a grande quantidade de parâmetros e de restrições, os algoritmos evolucionários (*Evolutionary Algorithms* - EA) são cada vez mais empregados na síntese de grades em fibra, obtendo resultados bastante satisfatórios. Este trabalho tem como objetivo principal desenvolver técnicas computacionais associadas à aplicação das EA na síntese de filtros ópticos com perfis de refletividade arbitrários. Para tanto, o *Genetic Algorithm* (GA) e o *Particle Swarm Optimization* (PSO) são enriquecidos com técnicas incorporadas da literatura e outras desenvolvidas pelo autor. As técnicas desenvolvidas classificam-se em métodos de decodificação (ou representação) e métodos de aceleração. Os métodos de decodificação são estratégias utilizadas na representação de uma FBG na forma de um vetor de busca, isto é, um cromossomo no caso do GA ou uma partícula no caso do PSO. Tais estratégias podem obter vantagens à EA através de uma significativa redução do espaço de soluções, seja limitando os intervalos dos parâmetros que o define, seja reduzindo o próprio número de parâmetros. Por outro lado, os métodos de aceleração atuam na amostragem espectral das curvas de refletividade calculadas durante a avaliação das soluções candidatas pela função objetivo. Melhorias dos algoritmos do GA e do PSO permitem a redução da quantidade de amostras exigidas pela função objetivo sem grandes impactos na qualidade das soluções finais. A avaliação das técnicas computacionais desenvolvidas foi feita por meio de simulações computacionais. Os resultados revelam que a combinação das técnicas de decodificação e de aceleração podem permitir fatores de ganho de desempenho computacional da ordem de 100 vezes, garantindo mesmo assim soluções satisfatórias.

PALAVRAS-CHAVE: Fibra Óptica, Filtros Ópticos, Grades de Bragg, GA, Algoritmo Genético, PSO, Enxame de Partículas.

Abstract

The Fiber Bragg Gratings (FBG) are flexible components. It has been largely deployed as main part of several devices used to assemble optical networks. Some applications can demand special devices made of FBG with unique characteristics. Special spectral response requirements are tied to Bragg gratings projects whose conception can be extremely difficult, demanding very specific computer techniques. Due the large amount of parameters and constrains, Evolutionary Algorithms (EA) are more and more applied for FBG synthesis, obtaining very satisfactory results. This work has as main objective the development of computational techniques associated to the application of EA on arbitrary reflectivity profile optical filters synthesis. This thesis uses Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) refined by techniques extracted from literature and by other ones developed by the author. All developed techniques can be classified as decodification (or representation) methods and computational acceleration methods. Decodification methods could be understood as specific ways how to represent a FBG as a search vector, i.e., a chromosome for GA or a particle for PSO. Such methods can achieve a significant reduction of the search space, by limiting all the intervals that defines the search space or by limiting the number of its dimensions. Acceleration methods operate on the way how the spectral sampling of reflectivity curves is made during the evaluation of each candidate solution by the objective function. Algorithm improvements on GA and PSO allow the reduction of the amount of samples demanded by the objective function without expressive impacts on the final solution quality. The developed techniques have been evaluated through computer simulations. The results show that the combination of decodification and acceleration techniques can allow to reach speed-up factors of about 100 times, nevertheless ensuring satisfactory solutions.

KEYWORDS: Optical fiber, Optical Filter, Bragg gratings, Genetic Algorithm, GA, Particle Swarm Optimization, PSO.

Sumário

Lista de Figuras	v
Glossário	xi
1 Introdução	1
1.1 Organização da tese	5
2 Grades de Bragg em fibra: análise e síntese	6
2.1 Modelagem de grades uniformes	8
2.2 Grades não uniformes apodizadas	10
2.2.1 Modelagem de grades não uniformes	12
2.2.2 Deslocamentos de fase	15
2.2.3 Grades com <i>chirp</i>	18
3 Síntese de grades utilizando algoritmo genético e de enxame de partículas	21
3.1 GA	21
3.1.1 GA binário	23
3.1.1.1 Operador de cruzamento	24
3.1.1.2 Operador de mutação	26
3.1.2 GA real	27
3.1.2.1 Operador de cruzamento	28
3.1.2.2 Operador de mutação	30
3.1.2.3 Observação das restrições	31
3.1.3 Operador de seleção	34
3.1.3.1 Seleção por roleta	35
3.1.3.2 Seleção por roleta com <i>ranking</i>	36

3.1.3.3	Seleção por torneio	36
3.1.3.4	Elitismo e a seleção elitista	38
3.1.4	Definições de parâmetros e auto-adaptação	41
3.1.4.1	O tamanho da população	41
3.1.4.2	Parâmetros de cruzamento	45
3.1.4.3	Parâmetros de mutação	48
3.1.4.4	A pressão seletiva	51
3.1.5	GA flexível	53
3.2	<i>Particle Swarm Optimization</i> (PSO)	56
3.2.1	Definição dos parâmetros	58
3.2.2	<i>Evolutionary</i> PSO	61
3.2.3	Topologias de cooperação entre partículas	63
3.2.4	Observação das restrições	63
3.2.5	PSO flexível	64
4	GA e PSO aplicados à síntese de FBG	69
4.1	Introdução	69
4.2	Decodificação	70
4.2.1	Decodificação normalizada	70
4.2.2	Decodificação com limites variáveis ao longo da FBG	72
4.2.2.1	Decodificação baseada em uma família de curvas - DFC	73
4.2.3	Decodificação empregando suavização	75
4.2.3.1	Decodificação suavizada restritiva - DSR	76
4.2.3.2	Decodificação suavizada pela média em movimento - DSMM	76
4.2.4	Decodificação usando interpolação	78
4.2.4.1	Decodificação usando interpolação linear - DIL	79
4.2.4.2	Decodificação usando spline quadrática - DSQ	79
4.2.4.3	Decodificação usando spline cúbica - DSC	80
4.2.5	Deslocamentos de fase	81
4.3	A função objetivo - FO	83
4.3.1	Amostragem progressiva - AP	86
4.3.2	Amostragem adaptativa - AA	89

4.3.3	Amostragem dinâmica - AD	93
4.3.4	Amostragem aleatória - AAL	93
4.3.4.1	Amostragem aleatória aplicada ao GA: o GA acelerado	94
4.3.4.2	Amostragem aleatória aplicada ao EPSO: o EPSO acelerado	97
5	Resultados	100
5.1	GA: estudo das técnicas de decodificação e de ace- leração	100
5.1.1	Parâmetros básicos do GA	100
5.1.1.1	GA: tamanho da população	101
5.1.1.2	Elitismo, Seleção e parâmetros correlatos	103
5.1.1.3	Mutação com desvio auto-adaptativo e o parâmetro τ_M	105
5.1.1.4	Mutação com desvio auto-adaptativo e cruzamento adaptativo com AGA	107
5.1.1.5	Alvo com tolerâncias	108
5.1.2	A decodificação suavizada	111
5.1.2.1	Decodificação suavizada para a síntese de filtros triangulares	115
5.1.3	A decodificação interpolada ou aproximada	122
5.1.3.1	Decodificação <i>spline</i> para a síntese de filtros triangulares	125
5.1.4	Aceleração do GA por meio da redução da quantidade de amostras da FO132	
5.2	PSO: estudo das técnicas de decodificação e de ace- leração	135
5.2.1	Parâmetros básicos do PSO	136
5.2.1.1	Tamanho do enxame	136
5.2.1.2	Pesos fixos, dinâmicos e auto-adaptativos	137
5.2.1.3	Definição dos parâmetros τ_P e P_Z	139
5.2.2	Técnicas de decodificação	141
5.2.2.1	Decodificação suavizada	142
5.2.2.2	Decodificação <i>spline</i>	143
5.2.3	Aceleração do EPSO por meio da redução da quantidade de amostras da FO	143
5.3	Comparação entre o GA e o EPSO	150
6	Conclusões	152

6.1	Publicações	154
6.1.1	Artigos publicados em periódicos	154
6.1.2	Artigos publicados em conferências	155

Lista de Figuras

1.1	Diagrama de um demultiplexador/multiplexador WDM <i>add-drop</i> utilizando uma FBG e um acoplador óptico.	2
1.2	Diagrama esquemático de um sensor óptico utilizando uma FBG como filtro-borda.	3
1.3	Espectros de um filtro borda TFBG (a) e outro baseado em uma FBG apodizada comum (b) sobrepostos com a potência óptica normalizada supostamente refletida pela FBG da cabeça sensora.	4
2.1	Diagrama da perturbação do índice de uma FBG uniforme.	8
2.2	Aspecto para apodizações cosseno-levantado baseadas em $\delta n(z)$ (a) e em ν (b).	11
2.3	Modelagem de uma FBG não uniforme através de sua subdivisão em M seções uniformes.	12
2.4	Comparação entre os espectros de uma grade uniforme e não uniforme com perturbação de índice baseada em δn	14
2.5	Comparação entre os espectros de uma grade uniforme e não uniforme com perturbação de índice baseada em ν	14
2.6	Espectros de uma grade uniforme e outra com apodização cosseno-levantado em ν , ambas com um deslocamento de fase de π a meio comprimento.	16
2.7	Efeito do ângulo de deslocamento sobre a posição da fenda no interior da banda de reflexão.	17
2.8	Efeito da posição do deslocamento sobre a profundidade da fenda no interior da banda de reflexão.	17
2.9	Efeito da ocorrência de mais de um deslocamento de π homogeneamente espaçados no interior da grade: dois deslocamentos em (a) e três em (b).	18
2.10	Efeito do <i>chirp</i> do alargamento na largura de banda de reflexão da FBG.	19

2.11	Efeito da posição do <i>chirp</i> do alargamento na largura de banda de reflexão da FBG.	20
3.1	Algoritmo Genético básico	22
3.2	Função Rastrigin bidimensional para $ x_k < 2$	24
3.3	Cruzamento de ponto único.	25
3.4	Cruzamento de dois pontos.	25
3.5	Cruzamento uniforme.	25
3.6	Mutação.	27
3.7	Função <i>int</i> para a decodificação de um número binário refletido.	28
3.8	Densidades para uma variável submetida a um desvio gaussiano sem e com truncagem simples.	32
3.9	Densidades para uma variável submetida a um desvio gaussiano sem e com truncagem utilizando re-execução.	33
3.10	Densidades para uma variável submetida a um desvio gaussiano sem e com truncagem utilizando a sobreposição de distribuição uniforme.	34
3.11	Algoritmo para a seleção Roulette-Wheel	35
3.12	Implementação da roleta desenrolada.	36
3.13	Seleção Roulette-Wheel com <i>ranking</i> linear	37
3.14	Seleção por torneio estocástico	38
3.15	Algoritmo Genético elitista básico	39
3.16	GA com seleção elitista	40
3.17	Algoritmo simplificado para o micro-GA	42
3.18	Algoritmo para o GA flexível	54
3.19	Algoritmo para o PSO simples	57
3.20	Articulação vetorial para o deslocamento das partículas no PSO.	58
3.21	Algoritmo para o EPSO simples	61
3.22	Três abordagens para a restrição dos parâmetros no PSO: fronteiras refletoras (a), absorvedoras (b) e transparentes (c).	64
3.23	Algoritmo para o PSO flexível	67

4.1	Exemplo onde três tipos de decodificação são comparadas: (a) empregando limites estáticos; (b) empregando curvas como limites e (c) empregando curvas como limites e apenas um escalar como parâmetro.	74
4.2	Exemplo onde várias curvas intermediárias, para diversos valores de $y_k(1)$, são mostradas entre as curvas limites $x_k^{min}(i)$ e $x_k^{max}(i)$	75
4.3	Comparação entre apodizações conseguidas para o mesmo cromossomo hipotético (ou equivalente PSO) utilizando a decodificação normal limitada por \mathbf{x}_k^{max} , suavização restritiva e suavização por média em movimento.	78
4.4	Comparação entre apodizações conseguidas para o mesmo cromossomo hipotético (ou equivalente PSO) utilizando a decodificação interpolada linear, ajustada quadrática e cúbica para limites constantes (a) e limites apodizados (b).	81
4.5	Função para a decodificação da subseção de cromossomo \mathbf{y}_5	82
4.6	Função de amostragem progressiva.	87
4.7	Curvas do valor objetivo em função do número de amostras para três tipos de FO diferentes: (a) para o erro RMS (FO_1), (b) para o erro quadrático cumulativo (FO_2) e (c) para o erro absoluto máximo (FO_3).	88
4.8	Três FBGs diferentes exemplo: (a) curvas de apodização para ν_i ; (b) curvas de refletividade.	89
4.9	Função de amostragem adaptativa.	90
4.10	Curvas do valor objetivo em função do número de amostras para três tipos de FO diferentes: (a) para o erro RMS (FO_1), (b) para o erro quadrático cumulativo (FO_2) e (c) para o erro absoluto máximo (FO_3).	92
4.11	Curvas de refletividade para FBG1, FBG2 e FBG3 obtidas através da amostragem adaptativa.	92
4.12	Algoritmo para o GA flexível acelerado.	96
4.13	Algoritmo para o EPSO flexível	98
5.1	Exemplo de uma curva para o valor de FO em função do número de gerações, sendo o “número de geração de convergência” (G_c) posto em destaque.	103
5.2	Variação do valor FO médio, desvio padrão (mostrado em barras) e valor mínimo em função de τ_M	107

5.3	Curva de refletividade para FBG sintetizada e alvo (a); curva de apodização obtida para a visibilidade de franjas ν (b).	109
5.4	Curva alvo respeitando intervalos de tolerância demarcado através de linhas tracejadas horizontais.	110
5.5	Curvas de refletividade (a) e de apodização da visibilidade de franjas ν (b) para uma FBG sintetizada utilizando o alvo original e para outra obtida com o alvo com tolerâncias.	111
5.6	Curvas de refletividade em dB (a) e curvas e perfis de apodização (b) para ν de grades obtidas com decodificação convencional e DSR ($p = 0.1$).	113
5.7	Curvas de refletividade em dB (a) e curvas e perfis de apodização (b) para ν de grades obtidas com DSR para $p = 0.02, 0.05$ e 0.1	114
5.8	Curvas de refletividade em dB (a) e curvas e perfis de apodização (b) para ν de grades obtidas com DSMM para $span = 2, 5$ e 10	114
5.9	Curvas de refletividade (a) e curvas e perfis de apodização para ν (b) e para λ_B (c) de grades obtidas com decodificação convencional e DSMM ($span = 5$). . .	117
5.10	Família de retas de possíveis apodizações de λ_B	118
5.11	Curvas de refletividade (a) e curvas e perfis de apodização para ν (b) e para λ_B (c) de grades obtidas com decodificação convencional e DSMM ($span = 2$) para ν . Ambas utilizando DFC para λ_B	120
5.12	Curvas de refletividade (a) e curvas e perfis de apodização para ν (b) e para λ_B (c) de grades obtidas com decodificação convencional e DSMM ($span = 10$) para λ_B . Ambas utilizando DFC para ν	121
5.13	Curvas de refletividade (a) e curvas e perfis de apodização para ν (b) de grades obtidas com decodificação suavizada cúbica, quadrática e linear.	124
5.14	Curvas de refletividade (a) e curvas e perfis de apodização para ν (b) de grades obtidas com DSC, DSQ e DIL.	125
5.15	Curvas de refletividade (a) e curvas de amplitude de perturbação do índice (b) e o perfil de λ_B (c).	127
5.16	Curvas de refletividade (a) e curvas de amplitude de perturbação do índice (b) e o perfil de λ_B (c).	128

5.17	Curvas de refletividade (a) e curvas de amplitude de perturbação do índice (b) e o perfil de λ_B (c).	129
5.18	Curvas de refletividade (a) e curvas de amplitude de perturbação do índice (b) e o perfil de λ_B (c).	130
5.19	Curvas de refletividade (a) e curvas de amplitude de perturbação do índice (b) e o perfil de λ_B (c).	131
5.20	Curvas de erro RMS em função do número de amostras (empregado pela FO) para o GA com amostragem fixa, aleatoriamente deslocada e para o GA acelerado: (a) curvas obtidas para o projeto refletor simples; (b) curvas obtidas para o projeto filtro rampa.	133
5.21	Curvas de refletividade obtidas para o GA empregando FO com apenas 1 amostra ($S_S = 100$ para o GA acelerado “C”): (a) curvas de refletividade para o projeto refletor simples; (b) curvas de refletividade para o projeto filtro rampa.	134
5.22	Curvas de ganho calculadas em função do tempo de processamento por geração (a) e em função do tempo de processamento total do GA.	135
5.23	Curva de <i>speed-up</i> efetivo em função de S_R para o GA acelerado em relação ao convencional utilizando 100 amostras fixas.	137
5.24	Curvas de erro RMS obtidas para o projeto de filtro refletor simples: (a) em função de P_Z para $\tau_P = 0.1$ e (b) em função de τ_P para $P_Z = 0.5$ (b).	141
5.25	Curvas de erro RMS obtidas para o projeto de filtro rampa: (a) em função de P_Z para $\tau_P = 0.05$ e (b) em função de τ_P para $P_Z = 0.25$ (b).	142
5.26	Curvas de refletividade em função do comprimento de onda (a) e de perfil de visibilidade de franjas ν em função do número da seção (b) para duas grades: uma obtida pelo EPSO sem decodificação suavizada e outra com DSR com $p = 0.1$	145
5.27	Curvas de refletividade em função do comprimento de onda (a) e de perfil de visibilidade de franjas ν em função do número da seção (b) para duas grades: uma obtida pelo EPSO sem decodificação suavizada e outra com DSR com $p = 0.1$	146
5.30	Curvas de erro RMS em função do número de amostras para três tipos de amostragem, para o projeto de filtro refletor (a) e projeto de filtro rampa (b).	146

5.31	Curva de refletividade para um filtro refletor um obtido pelo EPSO acelerado (a) e seu respectivo perfil de ν (b).	147
5.32	Curvas de refletividade para dois filtros rampa, um obtido pelo EPSO convencional e outro pelo EPSO acelerado (a) e os respectivos perfis de δn_{eff} (b) e de λ_B (c).	148
5.28	Curvas de refletividade em função do comprimento de onda (a) e de perfil de visibilidade de franjas ν para duas grades obtidas pelo EPSO empregando DSQ com 5 e 7 pontos mestres.	149
5.33	Curvas de ganho do EPSO acelerado em relação ao EPSO convencional: curvas obtidas em função do tempo de processamento médio de uma iteração (a); curvas obtidas em função do tempo de processamento total.	149
5.29	Curvas de refletividade em função do comprimento de onda (a), de perfil de perturbação δn_{eff} em função do número da seção (b) e perfil de λ_B para duas grades obtidas pelo EPSO empregando DSQ com 5 e 7 pontos mestres.	150

Glossário

AA	-	Amostragem Adaptativa
AAL	-	Amostragem Aleatória
AD	-	Amostragem Dinâmica
AGA	-	<i>Adaptive Genetic Algorithm</i>
AP	-	Amostragem Progressiva
APC	-	<i>Adaptive Parameter Control</i>
APSO	-	<i>Adaptive Particle Swarm Algorithm</i>
BB	-	<i>Building-Block</i>
BLX	-	<i>Blend Crossover</i>
CMAES	-	<i>Covariance Matrix Adapted Evolution Strategy</i>
DIL	-	Decodificação Interpolada Linear
DPC	-	<i>Dynamic Parameter Control</i>
DSC	-	Decodificação <i>Spline</i> Cúbica
DSMM	-	Decodificação Suavizada por Média Móvel
DSQ	-	Decodificação <i>Spline</i> Quadrática
DSR	-	Decodificação Suavizada Restritiva
DWDM	-	<i>Dense Wavelength-Division Multiplexing</i>
EA	-	<i>Evolutionary Algorithms</i>
EPSO	-	<i>Evolutionary Particle Swarm Algorithm</i>
ES	-	<i>Evolution Strategies</i>
FBG	-	<i>Fiber Bragg Gratings</i>

FO	-	Função Objetivo
FWHM	-	<i>Full Width at Half Maximum</i>
GA	-	<i>Genetic Algorithm</i>
GLM	-	<i>Gelfand-Levitan-Marchenko</i>
HPSO	-	<i>Hierarchical Particle Swarm Algorithm</i>
ITU-T	-	<i>International Telecommunications Union - Telecommunication standardization sector</i>
LASER	-	<i>Light Amplification by Stimulated Emission of Radiation</i>
LED	-	<i>Light Emitting Diode</i>
LP	-	<i>Layer Peeling</i>
LPG	-	<i>Long Period Gratings</i>
MPGA	-	<i>Multi-Population Genetic Algorithm</i>
PSO	-	<i>Particle Swarm Optimization</i>
RMS	-	<i>Root Mean Square</i>
RW	-	<i>Roulette-Wheel</i>
SAPC	-	<i>Self-Adaptive Parameter Control</i>
SBX	-	<i>Simulated Binary Crossover</i>
SSGA	-	<i>Stead State Genetic Algorithm</i>
TFBG	-	<i>Triangular Fiber Bragg Grating</i>
UV	-	Ultravioleta
WDM	-	<i>Wavelength-Division Multiplexing</i>
XHC	-	<i>Crossover Hill-Climbing</i>

Capítulo 1

Introdução

A partir dos anos noventa do século passado, o mundo assistiu a disseminação das redes ópticas auxiliada por uma nova geração de componentes. Os amplificadores ópticos dispensavam a transição do sinal entre os domínios elétrico e óptico, tornando os regeneradores optoeletrônicos praticamente obsoletos e aumentando consideravelmente o alcance das redes ópticas.

Uma vez que a atenuação não representava mais uma grande limitação ao avanço das redes de telecomunicação ópticas, outros fenômenos, como a dispersão cromática, tomaram de súbito a atenção dos pesquisadores. Ao mesmo tempo havia uma demanda inexoravelmente crescente por largura de banda, que culminou na disseminação dos sistemas multiplexados em comprimento de onda (*Wavelength-Division Multiplexing* – WDM). Para ambos, o combate à dispersão e a viabilização dos sistemas multiplexados, são utilizados dispositivos baseados em grades de Bragg em fibra (*Fiber Bragg Gratings* – FBG).

Uma FBG é um tipo de refletor Bragg construído no interior de uma fibra óptica, capaz de refletir uma faixa de comprimentos de onda, conservando-se relativamente transparente para o restante do espectro. Seu princípio baseia-se na variação periódica do índice de refração do núcleo da fibra óptica, cujas propriedades podem conferir à FBG um coeficiente de reflexão função do comprimento de onda incidente. As FBG possuem algumas vantagens em comparação a outras tecnologias, como os refletores Bragg planares: possuem a mesma geometria que a fibra óptica, portanto pequenas perdas por inserção nos circuitos ópticos; alta seletividade ou qualidade dos filtros e relativo baixo custo.

A Fig. 1.1 exibe o esquema de um dispositivo de remoção e de adição (*add-drop*) de portadoras ópticas, tipicamente utilizado em redes WDM. A seletividade da FBG é explorada de modo a refletir apenas λ_3 de volta ao circulador óptico **C**. Após a FBG, a portadora λ_3 é reinserida no circuito através do acoplador óptico **A**.

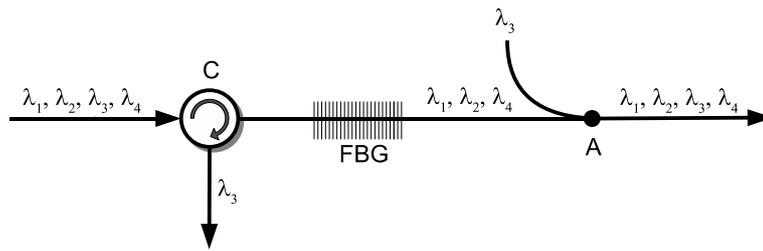


Figura 1.1: Diagrama de um demultiplexador/multiplexador WDM *add-drop* utilizando uma FBG e um acoplador óptico.

Além das redes de comunicação, as FBG são também aplicadas em redes de sensores ópticos: (1) como cabeças ou elementos sensores, uma vez que as FBG são sensíveis a várias grandezas físicas capazes de afetar a geometria ou os valores de índice de refração da fibra. Mudanças de temperatura ou deformações mecânicas podem deslocar a banda de reflexão da FBG de uma forma mensurável e proporcional. (2) integrando os sistemas de interrogação da rede de sensores. Os sistemas de interrogação são constituídos por um dispositivo de excitação, um laser ou LED, e um aparato de leitura. O excitador ilumina os elementos sensores dispostos em série ao longo da mesma fibra óptica os quais refletem a luz de volta ao aparato de leitura, para que este realize a avaliação dos deslocamentos ou deformações espectrais. Porém, as técnicas tradicionais de avaliação, como as que fazem usos de analisadores de espectro ópticos, podem ser realmente dispendiosas. É necessário recorrer a outros métodos de leitura capazes de converter a informação relacionada aos deslocamentos espectrais em grandezas de processamento simples e econômico. Para tanto existem os filtros de borda, freqüentemente não mais que uma FBG especialmente designada de modo a converter deslocamentos espectrais em flutuações de potência óptica, cuja coleta pode ser realizada mediante simples fotodetectores. A Fig. 1.2 exibe um diagrama esquemático para um sensor utilizando filtro de borda no aparato de interrogação. A fonte banda larga **F** ilumina o sensor. A luz refletida é redirecionada pelo circulador **C** para o filtro de borda que divide a luz incidente em duas componentes: a refletida, detectada pelo fotodetector **R1**; e a transmitida, detectada pelo fotodetector **R2**. A relação entre as potências ópticas registradas entre os dois detectores fornece informação suficiente a cerca do deslocamento espectral sofrido pelo sensor FBG.

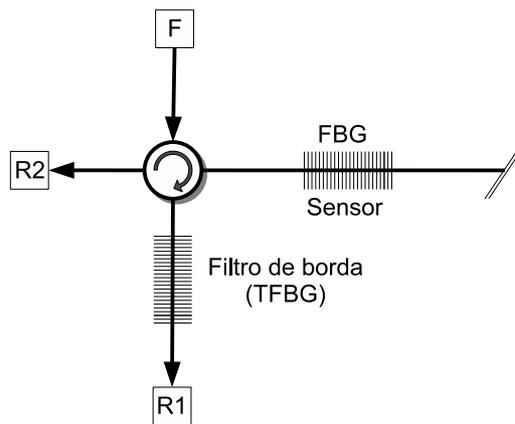


Figura 1.2: Diagrama esquemático de um sensor óptico utilizando uma FBG como filtro-borda.

Em [1] é proposto o uso de FBG triangulares (*Triangular FBG* – TFBG) como filtros de borda. As TFBG possuem uma curva de refletividade que varia linearmente no interior de uma faixa de comprimentos de onda. De acordo com [1], esta característica torna as TFBG superiores aos filtros de borda convencionais pois lhes confere maior sensibilidade, maior imunidade em relação a eventuais flutuações de potência dos excitadores e largura de banda de operação. A Fig. 1.3 mostra os espectros de refletividade de um filtro borda TFBG (a) e outra grade apodizada comum (b) sobrepostos com um suposto espectro de potência normalizado refletido pela cabeça sensora. Em vermelho destaca-se a parte transmitida dos sinais. É evidente a superioridade de uma TFBG nesta aplicação, que permite excelentes medidas em uma faixa espectral muito mais larga. As TFBG são um bom exemplo da flexibilidade das FBG, que possuem muitos parâmetros relacionados ao padrão de variação do índice de refração cujo ajuste permite obter filtros com espectros de refletividade apropriados para quaisquer necessidades e aplicações. Infelizmente, a determinação destes parâmetros não é um problema trivial. A síntese, isto é, a determinação das características de construção de uma FBG dadas certas especificações do seu espectro, é um campo fértil para discussões e inúmeras metodologias têm sido propostas. Para grades fracas cuja refletividade é pequena, o problema de síntese pode ser reduzido à transformada de Fourier do coeficiente de reflexão. Essa abordagem foi estendida por Winick e Roman [2] de modo a permitir a síntese de grades com elevada refletividade, embora de forma ainda aproximada. As técnicas baseadas em *Layer Peeling* (LP) [3], por outro lado, são capazes de obter soluções arbitrariamente precisas, porém tem a desvantagem de obter longos e complicados perfis de índice de refração, resultando projetos cuja realização depende de aparatos de fabricação especiais e dispendiosos. Para tentar atender as restrições associadas às técnicas de fabricação, Askanes *et al* [4] propôs um método híbrido que combina busca local com um algoritmo LP.

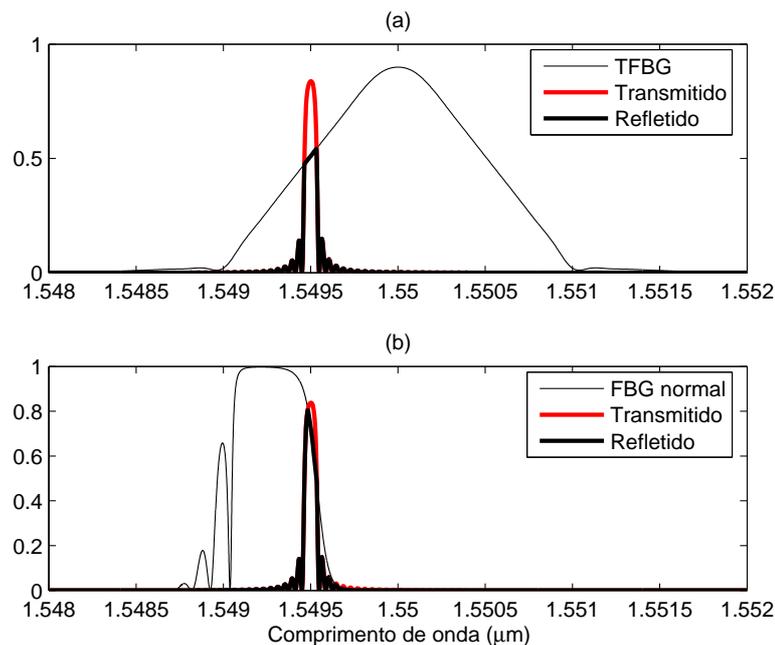


Figura 1.3: Espectros de um filtro borda TFBG (a) e outro baseado em uma FBG apodizada comum (b) sobrepostos com a potência óptica normalizada supostamente refletida pela FBG da cabeça sensora.

Por outro lado, *Evolutionary Algorithms* (EA), como o algoritmo genético (*Genetic Algorithm* – GA) [5] e a otimização por enxames de partículas (*Particle Swarm Optimization* – PSO) [6], são capazes de alcançar bons resultados mesmo quando restrições adicionais são impostas ao processo de otimização com o objetivo de garantir a viabilidade dos projetos. De fato, o uso dos EA para síntese de FBG é bastante recorrente na literatura. Em [7], um dos artigos pioneiros no uso do GA para síntese de FBG, o processo de otimização utiliza restrições associadas tanto ao módulo quanto à fase do coeficiente de reflexão, resultando em filtros ópticos de dispersão aproximadamente nula. Em [8], um artigo bem mais recente, a síntese de filtros FBG de dispersão nula é novamente abordada, porém utilizando um outro tipo de EA em muitos aspectos similar ao GA, uma estratégia evolutiva adaptada por matriz de covariância (*Covariance Matrix Adapted Evolution Strategy* – CMAES). Este mesmo tipo de projeto é visitado pela terceira vez em [9], desta vez utilizando o PSO. Em [1] o GA é a heurística empregada na síntese de grades triangulares. Em [10] a síntese das TFBG é realizada através do CMAES, para o qual é demonstrando a superioridade frente ao PSO clássico.

Esta Tese tem como objetivo principal desenvolver e aplicar técnicas computacionais associadas à aplicação das heurísticas GA e PSO na síntese de filtros ópticos com perfis de refletividade arbitrários. Conciliando os conhecimentos sobre a análise computacional das FBG

com as restrições dos processos de manufatura, este trabalho procura desenvolver métodos e algoritmos especializados, permitindo às heurísticas um alto ganho de eficiência tanto na qualidade das soluções finais quanto em economia de recursos computacionais.

1.1 Organização da tese

Deixando à parte a introdução e as conclusões, esta Tese organiza-se basicamente em 4 capítulos:

- O Capítulo 2 explora o modelo e o método matemático matricial utilizado para a análise das grade de Bragg, que será doravante empregado pelas heurísticas GA e PSO;
- O Capítulo 3 apresenta o GA e o PSO, propondo também versões modificadas destes algoritmos;
- O Capítulo 4 apresenta quase todas as principais contribuições da Tese:
 - Técnicas de amostragem progressiva e adaptativa;
 - Técnicas de decodificação suavizadas;
 - Técnicas de decodificação baseadas em famílias de curvas;
 - Utilização de limites variáveis em função do comprimento da FBG. Esta técnica de decodificação será freqüentemente referenciada como “curvas limites”;
 - Técnicas de decodificação empregando interpolação ou aproximações *spline*;
 - Técnicas de aceleração do GA e PSO empregando redução de amostragem;
- O Capítulo 5 aplica o GA e o PSO propostos no Capítulo 3, explorando ainda quase todas as técnicas apresentadas no Capítulo 4.

Capítulo 2

Grades de Bragg em fibra: análise e síntese

Uma grade em fibra é um trecho de fibra óptica que exhibe, em seu núcleo, uma perturbação periódica do índice de refração. Esta perturbação confere à grade a capacidade de refletir ou transmitir a luz de forma mais ou menos intensa em função do comprimento de onda, característica largamente aproveitada em diversos dispositivos empregados em redes de comunicações e de sensores ópticos.

As grades em fibra podem ser classificadas em grades de reflexão e em grades de transmissão. As grades de reflexão, grades de Bragg em fibra (*Fiber Bragg Gratings* – FBG) ou de período curto, são aquelas cujo acoplamento ocorre entre os modos do núcleo que se propagam em direções opostas; por outro lado, as grades de transmissão, também chamadas de grades de período longo (*Long Period Gratings* – LPG), são aquelas cujo acoplamento se dá entre modos do núcleo e da casca que propagam-se na mesma direção. Esta Tese se deterá apenas no estudo das FBG.

As grades em fibra são fabricadas aproveitando a fotossensibilidade das fibras dopadas em relação à luz ultravioleta, capaz de provocar uma permanente mudança no índice de refração do núcleo da fibra óptica. Geralmente o dopante utilizado nas fibras fotossensíveis é o Germânio, um elemento capaz de tomar o lugar do silício nas ligações com o Oxigênio, ocasionando defeitos estruturais que manifestam-se como alterações das propriedades elétricas da sílica. Esses defeitos aumentam ou modificam-se proporcionalmente à incidência da radiação apropriada (normalmente ultravioleta – UV). Muito embora a fotossensibilidade tenha sido descoberta a mais de 30 anos, os seus princípios ainda não foram totalmente esclarecidos até hoje. Existem algumas teorias a respeito, mas é de consenso geral a atribuição da

fotossensibilidade aos defeitos de inclusão da sílica [11].

A fotossensibilidade da fibra óptica pode ser melhorada através da hidrogenização, um processo que permite a inclusão de átomos de Hidrogênio no interior da sílica. Através da irradiação por UV, o Hidrogênio interfere com as ligações Silício-Oxigênio-Germânio, reagindo com o Oxigênio e formando OH. Portanto, os defeitos estruturais se multiplicam, intensificando a fotossensibilidade da fibra. A dopagem com apenas Germânio permite perturbações do índice de refração da ordem de 10^{-5} até 10^{-4} . Entretanto a hidrogenização permite perturbações superiores, da ordem de 10^{-3} até 10^{-2} . Este processo, entretanto, é realizado através da imersão da fibra óptica em uma atmosfera de Hidrogênio a altas pressões por até uma semana. Este período pode ser reduzido para algo próximo de dois dias aquecendo a atmosfera de Hidrogênio a 150°C . Este processo tem como principal vantagem a capacidade de tornar a fibra óptica padrão fotossensível. Dentre algumas desvantagens deste processo destaca-se o risco de acidentes, dado que o Hidrogênio é um gás inflamável e precisa ser mantido aquecido e a altas pressões, além da possibilidade do acúmulo excessivo de OH na fibra, aumentando a absorção da luz na região da segunda janela [11].

De posse de uma fibra óptica fotossensível, resta produzir padrões de perturbação de índice de refração com o auxílio de um feixe de luz ultravioleta. Para tanto, a luz UV não deve irradiar a fibra de forma equânime em toda extensão fotossensível, mas seguindo o padrão periódico desejado, uma vez que a variação induzida do índice de refração é proporcional à intensidade luminosa e ao tempo de exposição.

O processo de transferência do padrão luminoso UV para o índice de refração da fibra é comumente chamado de gravação ou inscrição. O primeiro processo de gravação foi demonstrado por Hill em 1978 [12], que iluminou uma fibra com laser visível de Argônio, gerando um padrão de onda estacionária no seu núcleo. Em 1989, Meltz *et al* [13] demonstraram uma técnica que ainda é empregada até hoje, a gravação holográfica lateral utilizando um padrão interferométrico conseguido através de dois feixes de luz UV. Variantes contemporâneas do processo interferométrico podem utilizar prismas para dividir o feixe de luz.

Outra técnica bastante interessante é a inscrição direta ou ponto a ponto. Neste caso, um laser UV é focado em uma região diminuta da fibra óptica, correspondendo ao período de perturbação desejado para o índice de refração. Então a fibra é cuidadosamente deslocada deste mesmo período para que o laser torne a ser ativado em outra região contígua da fibra. O tempo de exposição da fibra define a intensidade da perturbação do índice. Este processo permite total flexibilidade de projeto, sendo teoricamente capaz de gravar grades arbitrariamente longas, desde que questões críticas relacionadas ao controle de deslocamento da fibra possam ser satisfatoriamente solucionadas [14].

A técnica de gravação considerada como mais efetiva é feita através de máscaras de fase. Uma máscara é um elemento difrator que, inserido entre a fonte luminosa e a fibra, gera uma sombra cujo padrão de intensidades acompanha o padrão da máscara. Padrões complexos de periodicidade variável são possíveis, desde que se disponha de uma máscara correspondente. Dado que a técnica interferométrica não permite o mesmo, essa técnica de gravação é uma importante alternativa na manufatura de grades com *chirp* [15].

2.1 Modelagem de grades uniformes

A perturbação periódica do índice de refração gravada no núcleo de uma FBG pode ser modelada da seguinte forma [16]:

$$n(z) = n_{eff} + \delta n \left[1 + \nu \cos \left(\frac{2\pi z}{\Lambda} \right) \right] \quad (2.1)$$

Onde n_{eff} representa o índice efetivo do núcleo da fibra, δn representa o incremento do índice de refração médio, ν é a visibilidade de franjas ou razão de modulação, z representa a posição da grade e Λ o período de modulação. A Fig. 2.1 ilustra o que os parâmetros de (2.1) representam no perfil de índices de refração periódico de uma FBG de comprimento L , retratada como uma curva senóide. Também, nesta figura, uma origem para o eixo z é adotada seguindo a convenção da esquerda para a direita.

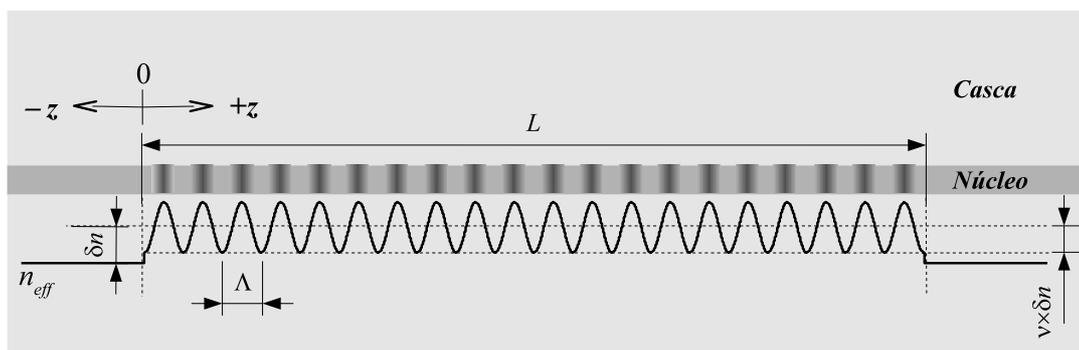


Figura 2.1: Diagrama da perturbação do índice de uma FBG uniforme.

A perturbação do índice de refração no interior do núcleo da fibra óptica gera, por reflexão, um outro modo contra-propagante (sentido $z-$) a partir do modo principal (sentido $z+$). A relação que se estabelece entre os dois modos pode ser descrita com o auxílio da teoria dos modos acoplados. A formulação base desta teoria é apresentada de forma aprofundada em

[17] e desenvolvida em [18], assumindo um formalismo matricial de acordo com o qual uma grade uniforme pode ser representada através de uma matriz de transferência 2x2 \mathbf{F} :

$$\begin{bmatrix} R_L \\ S_L \end{bmatrix} = \mathbf{F} \times \begin{bmatrix} R_0 \\ S_0 \end{bmatrix}, \quad (2.2)$$

onde R_0 e S_0 representam respectivamente as amplitudes dos campos para os modos propagantes e contrapropagantes em $z = 0$; R_L e S_L representam respectivamente as amplitudes dos campos para os modos propagantes e contrapropagantes em $z = L$; \mathbf{F} representa a matriz de transferência para a grade uniforme de comprimento L [16]:

$$\mathbf{F} = \gamma^{-1} \begin{bmatrix} \gamma \cosh(\gamma L) - j\hat{\sigma} \sinh(\gamma L) & -j\kappa \sinh(\gamma L) \\ j\kappa \sinh(\gamma L) & \gamma \cosh(\gamma L) + j\hat{\sigma} \sinh(\gamma L) \end{bmatrix}. \quad (2.3)$$

Onde $j = \sqrt{-1}$ e os parâmetros κ , $\hat{\sigma}$ e γ são dados respectivamente por:

$$\kappa = \frac{\pi}{\lambda} \nu \delta n, \quad (2.4)$$

$$\hat{\sigma} = \pi \left[\frac{2}{\lambda} (\delta n + n_{eff}) - \frac{1}{\Lambda} \right] \quad (2.5)$$

e

$$\gamma = \sqrt{\kappa^2 - \hat{\sigma}^2}. \quad (2.6)$$

A partir de (2.2), fazendo $R_0 = 1$ e $S_0 = 0$, resultará $R_L = \rho$ e $S_L = \tau$, respectivamente os coeficientes de reflexão e transmissão. Resolvendo o sistema de duas equações decorrente, pode-se obter o coeficiente de reflexão ρ em função de κ , $\hat{\sigma}$ e γ [16]:

$$\rho = \frac{-\kappa \sinh(\gamma L)}{\hat{\sigma} \sinh(\gamma L) + j\gamma \cosh(\gamma L)} \quad (2.7)$$

A refletividade, isto é, o coeficiente de reflexão em termos de potência, pode ser obtido a partir de $r = |\rho|^2$ como:

$$r = \frac{\sinh^2(\gamma L)}{\cosh^2(\gamma L) - \hat{\sigma}^2/\kappa^2} \quad (2.8)$$

Através do estudo da derivada de r em função de $\hat{\sigma}$, conclui-se que a refletividade atinge seu máximo para $\hat{\sigma} = 0$. Condição esta que ocorre para o comprimento de onda λ_{max} :

$$\lambda_{max} = \left(1 + \frac{\delta n}{n_{eff}}\right) \lambda_D \quad (2.9)$$

onde $\lambda_D = 2n_{eff}\Lambda$.

2.2 Grades não uniformes apodizadas

Graças a sua construção, imersa na fibra óptica, as FBG possuem uma perda por inserção realmente muito pequena. Essa é uma das principais características que as tornam importantes junto aos sistemas WDM. De fato, as FBG podem ser encontradas no cerne das principais arquiteturas de multiplexação e demultiplexação DWDM (*Dense* – WDM). Entretanto, uma outra demanda importante destes sistemas é a separação entre os canais. A FBG uniforme não consegue atender às especificações ITU–T [19] conservando, ao mesmo tempo, altas refletividades no interior da banda. A razão disso se deve ao perfil de índices que se inicia e termina abruptamente, gerando harmônicas indesejáveis que alargam demasiadamente o espectro de refletividade, portanto invadindo outras faixas espectrais de outros canais DWDM contíguos. Para combater esse efeito recorre-se à apodização do perfil de índice de refração.

Uma FBG apodizada é uma grade não uniforme, isto é, que possui pelo menos um dentre os parâmetros δn , ν e Λ em função da posição da grade z . Nas grades apodizadas, entretanto, em geral apenas o parâmetro δn ou ν varia lentamente em função de z com o objetivo de suavizar os limites da FBG, eliminando o efeito de confinamento. Em geral o formato das funções $\delta n(z)$ ou $\nu(z)$ segue um perfil “fusiforme”, assumindo valores de amplitude mínimos para $z = 0$ e $z = L$, e máximos para $z = L/2$. A Fig. 2.2 mostra exemplos típicos de apodizações com o formato dado pela função cosseno-levantado.

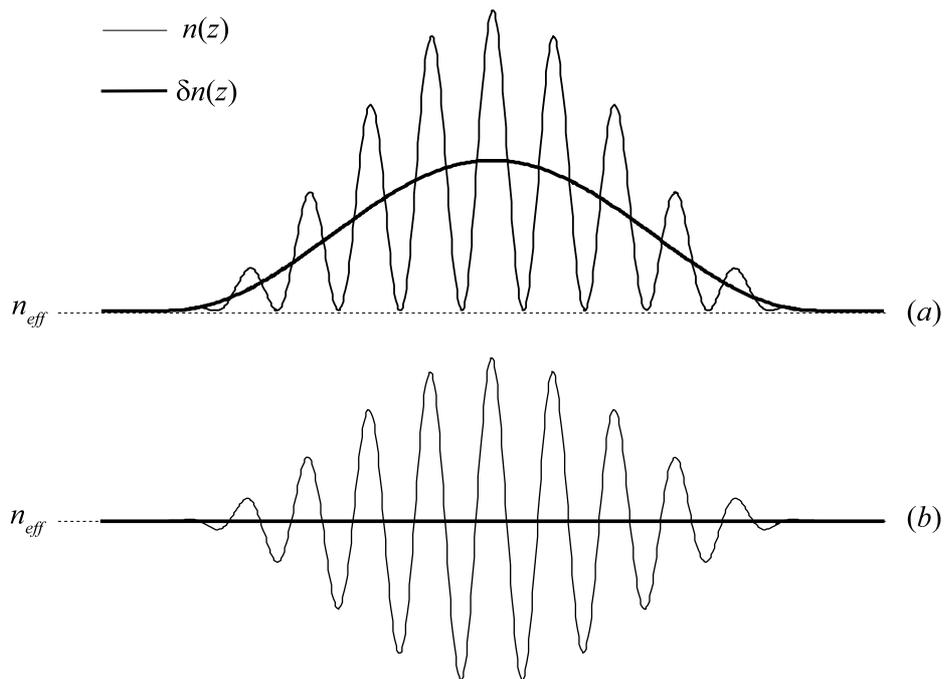


Figura 2.2: Aspecto para apodizações cosseno-levantado baseadas em $\delta n(z)$ (a) e em ν (b).

A Fig. 2.2 (a) é uma apodização baseada em $\delta n(z)$ conservando a visibilidade de franjas constante $\nu = 1$. Por outro lado, em (b) é mostrada o aspecto de uma apodização baseada na visibilidade de franjas, conservando constante $\delta n(z) \rightarrow 0$ [16]. O caso (b) apresenta uma grande vantagem em comparação ao caso (a). Para explicá-la, é necessário lembrar que o processo de gravação das FBG em fibra é função de vários parâmetros. A máxima perturbação do índice depende da sensibilidade da fibra, do tempo de exposição, intensidade da fonte laser UV entre outros fatores. Vários destes parâmetros reais são aproximados e sempre acompanhados de certa percentagem de incerteza. Como resultado, uma gravação de FBG nunca poderá ser repetida exatamente da mesma forma e, para o caso (a), o máximo valor de δn certamente será variável. De acordo com (2.9), o comprimento de onda de máximo λ_{max} é função de δn , o que significa que quaisquer incertezas associadas a este parâmetro serão traduzidas em um deslocamento espectral imprevisível. Por outro lado, para o caso (b), o incremento médio do índice é nulo. Nesta condição λ_{max} será sempre igual a λ_D , havendo muito mais controle sobre o posicionamento da banda de reflexão no espectro.

Contudo, a fabricação de uma FBG com a apodização baseada em ν é mais complexa do que a baseada em δn . Um perfil do tipo mostrado em Fig. 2.2 (a) pode ser conseguido através de uma exposição simples da fibra usando um método interferométrico e barreiras semi-transparentes para reduzir a incidência UV nos extremos da FBG. Por outro lado, o perfil como o (b) depende de exposições duplas no qual o nível “dc” do índice de refração

nos extremos da grade são reforçados antes da gravação do perfil “ac”. Uma outra técnica utiliza uma máscara de fase com a fibra colocada em oscilação longitudinal (conservando a fibra a uma distância fixa da máscara de fase) e com exposição janelada, capaz de comutar entre exposição “ac” e “dc” em função da amplitude de oscilação da fibra. Por fim, pode ser necessário empregar máscaras apodizadas, que são muito mais dispendiosas do que as máscaras para grades uniformes simples.

2.2.1 Modelagem de grades não uniformes

Uma grade não uniforme, tal como uma grade apodizada, pode ser modelada como uma série de M grades uniformes curtas (seções) concatenadas, cada qual com uma combinação diferente de parâmetros. A Fig. 2.3 ilustra a subdivisão da FBG apodizada em várias grades uniformes, cada uma delas definida através de δn_k , ν_k , Λ_k e Δz_k , onde o subscrito k indica o número da grade uniforme e Δz_k representa o comprimento da seção k .

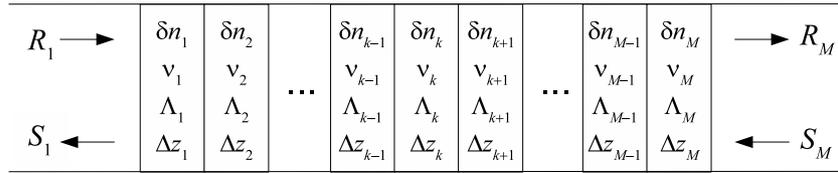


Figura 2.3: Modelagem de uma FBG não uniforme através de sua subdivisão em M seções uniformes.

A matriz de transferência associada à seção uniforme k , \mathbf{F}_k , pode ser obtida através de (2.3) substituindo L por Δz_k :

$$\mathbf{F}_k = \gamma_k^{-1} \begin{bmatrix} \gamma_k \cosh(\gamma_k \Delta z_k) - j \hat{\sigma}_k \sinh(\gamma_k \Delta z_k) & -j \kappa_k \sinh(\gamma_k \Delta z_k) \\ j \kappa_k \sinh(\gamma_k \Delta z_k) & \gamma_k \cosh(\gamma_k \Delta z_k) + j \hat{\sigma}_k \sinh(\gamma_k \Delta z_k) \end{bmatrix}. \quad (2.10)$$

Onde $\gamma_k = \sqrt{\kappa_k^2 - \hat{\sigma}_k^2}$, $\kappa = \pi \nu_k \delta n_k / \lambda$ e $\hat{\sigma}_k = \pi [2\lambda^{-1}(\delta n_k + n_{eff}) - \Lambda_k^{-1}]$.

A matriz de transferência de toda a grade apodizada é dada pelo produto das matrizes de transferência das seções uniformes:

$$\mathbf{F}_T = \mathbf{F}_M \times \mathbf{F}_{M-1} \dots \times \mathbf{F}_{k+1} \times \mathbf{F}_k \times \mathbf{F}_{k-1} \dots \times \mathbf{F}_2 \times \mathbf{F}_1. \quad (2.11)$$

As amplitudes R_M e S_M são dadas em função de R_1 , S_1 e da matriz de transferência total \mathbf{F}_T :

$$\begin{bmatrix} R_M \\ S_M \end{bmatrix} = \mathbf{F}_T \times \begin{bmatrix} R_1 \\ S_1 \end{bmatrix}; \mathbf{F}_T = \begin{bmatrix} \mathbf{F}_{T11} & \mathbf{F}_{T12} \\ \mathbf{F}_{T21} & \mathbf{F}_{T22} \end{bmatrix}. \quad (2.12)$$

Considerando $R_1 = 1$, $S_1 = \rho$, $R_M = \tau$ e $S_M = 0$, e resolvendo o sistema de duas equações decorrente de (2.12), chega-se ao coeficiente de reflexão $\rho = -\mathbf{F}_{T21}/\mathbf{F}_{T22}$ e ao coeficiente de transmissão $\tau = \det(\mathbf{F}_T)/\mathbf{F}_{T22}$, onde \det é uma função que calcula o determinante do argumento. O coeficiente de reflexão em termos de potência, a refletividade, pode ser obtido a partir de ρ como $r = |\rho|^2$.

A Fig. 2.4 compara as curvas de refletividade de uma grade uniforme com outra não uniforme e apodizada em δn de acordo com a função cosseno-levantado, com valor máximo de δn igual a 4×10^{-4} . A não ser pela apodização, ambas as grades possuem todos os outros parâmetros iguais: $L = 1\text{cm}$, $\lambda_D = 1550\text{nm}$, $\nu = 1$, $n_{eff} = 1.45$. Para a determinação do espectro da grade não uniforme, esta foi dividida em 100 seções uniformes com comprimentos Δz iguais a $100\mu\text{m}$ ($\Delta z > 187\Lambda$). De acordo com [18], o comprimento da seção Δz deve ser muito maior que o período Λ . Em [16] recomenda-se que a razão entre Δz e Λ não seja inferior a 100.

A Fig. 2.5 compara as curvas de refletividade de uma grade uniforme com outra não uniforme. Mas desta vez a figura explora a perturbação de índice de refração baseada na visibilidade de franjas ν , de modo que $\nu \times \delta n$ alcance no máximo 4×10^{-4} e $\delta n \rightarrow 0$ ($\delta n = 10^{-12}$). Das figuras 2.4 e 2.5 é importante observar que os espectros das grades apodizadas exibem lóbulos laterais muito menores do que os observados nas grades uniformes. Também que a apodização baseada exclusivamente em δn provoca o deslocamento das bandas de máxima reflexão para além de λ_D .

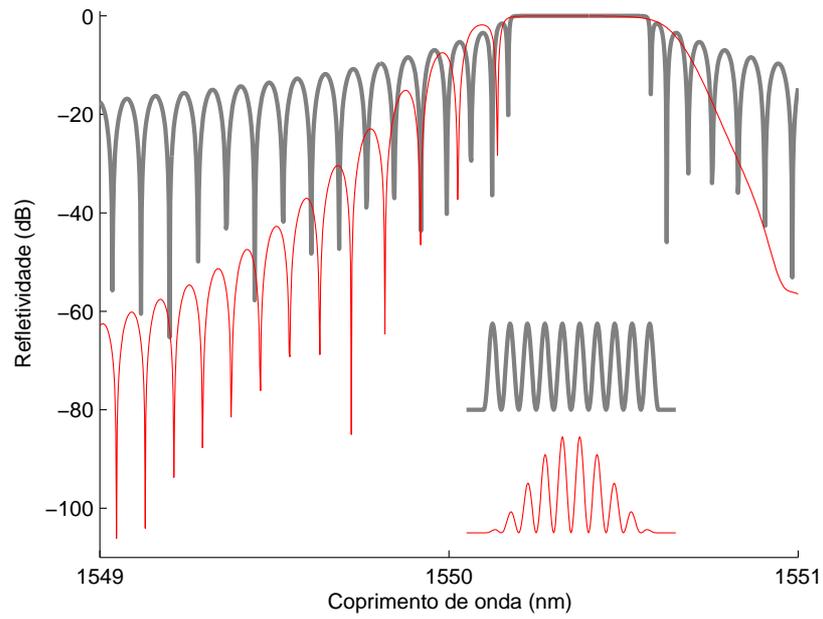


Figura 2.4: Comparação entre os espectros de uma grade uniforme e não uniforme com perturbação de índice baseada em δn .

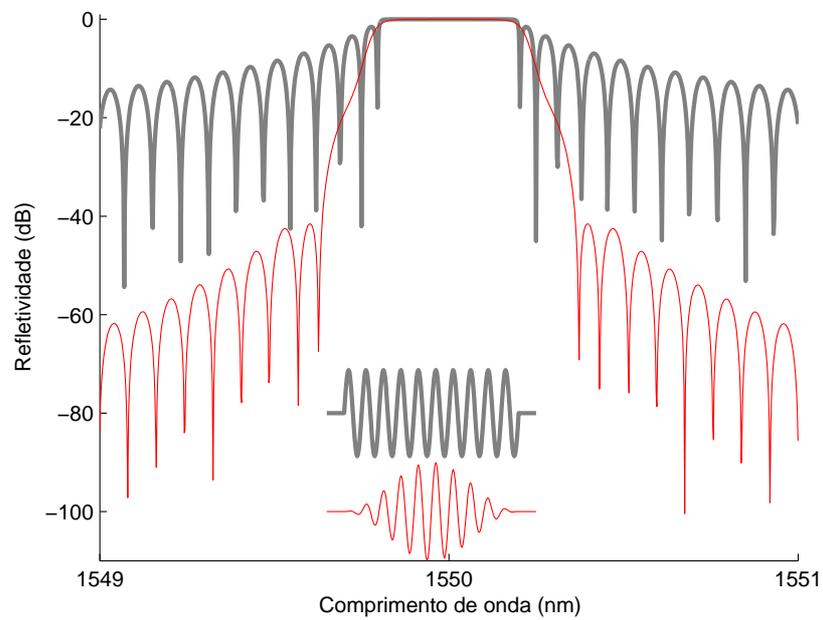


Figura 2.5: Comparação entre os espectros de uma grade uniforme e não uniforme com perturbação de índice baseada em ν .

2.2.2 Deslocamentos de fase

Um deslocamento de fase é uma descontinuidade do perfil periódico de índice de refração. Seu efeito no espectro de refletividade da FBG é de abrir uma fenda (*notch*) de transmissão na banda de reflexão. A largura da fenda de transmissão tende a ser relativamente pequena, chegando a valores tais como 10 picômetros, e não raro menores. Esta característica interessante favorece às FBG com deslocamento de fase inúmeras aplicações em redes de telecomunicações ópticas, em lasers semicondutores, sensoriamento óptico entre outras [20], [21].

É possível fabricar grades com deslocamentos de fase através de todos os principais métodos: interferométrico, direto ou através de máscaras de fase. Entretanto o método interferométrico é limitado quanto ao número de deslocamentos possíveis e também quanto valor do deslocamento, embora a superposição de grades permita obter o efeito de deslocamento de fase com alguns graus de liberdade, de acordo com a teoria acerca das grades de Moiré [22].

O deslocamento de fase pode ser facilmente incorporado à formulação matricial empregada com as grades não uniformes [18]. Considere um deslocamento de ϕ radianos em alguma posição z da FBG, entre as matrizes \mathbf{F}_k e \mathbf{F}_{k+1} . A matriz de transferência que modela este deslocamento é dado por:

$$\mathbf{F}_{\phi k} = \begin{bmatrix} \exp(-j\phi/2) & 0 \\ 0 & \exp(j\phi/2) \end{bmatrix}. \quad (2.13)$$

A expressão para \mathbf{F}_T (2.11) pode ser reescrita como:

$$\mathbf{F}_T = \mathbf{F}_M \times \mathbf{F}_{M-1} \dots \times \mathbf{F}_{k+1} \times \mathbf{F}_{\phi k} \times \mathbf{F}_k \times \mathbf{F}_{k-1} \dots \times \mathbf{F}_2 \times \mathbf{F}_1. \quad (2.14)$$

A matriz $\mathbf{F}_{\phi k}$ também pode ser utilizada para representar espaçamentos em branco, isto é, trechos desprovidos de perturbações no índice de refração no interior das grades, tal como ocorre nas grades amostradas [16]. Um espaçamento Δz_0 produz o mesmo efeito de um deslocamento de fase $\phi = 2\pi\Delta z_0/\Lambda$, onde Λ representa o período médio de perturbação dos trechos de grade antes e/ou depois do espaçamento em branco.

A Fig. 2.6 exhibe os espectros de duas grades com deslocamentos de fase. Ambas possuem o mesmo comprimento de 1cm e deslocamentos de π radianos em $z = 0.5\text{cm}$. Diferenciam-se pela apodização, sendo a primeira uniforme, com $\delta n \rightarrow 0$ e $\delta n \times \nu = 2 \times 10^{-4}$, e a segunda apodizada cosseno-levantado, com $\delta n \rightarrow 0$ e máxima amplitude de perturbação do índice $\delta n \times \nu = 4 \times 10^{-4}$. A grade apodizada foi modelada em 100 seções uniformes de

comprimentos iguais. As larguras FWHM (*Full Width at Half Maximum*) registradas para as fendas de transmissão foram de aproximadamente 7.6 e 14.2pm para a FBG uniforme e apodizada respectivamente.

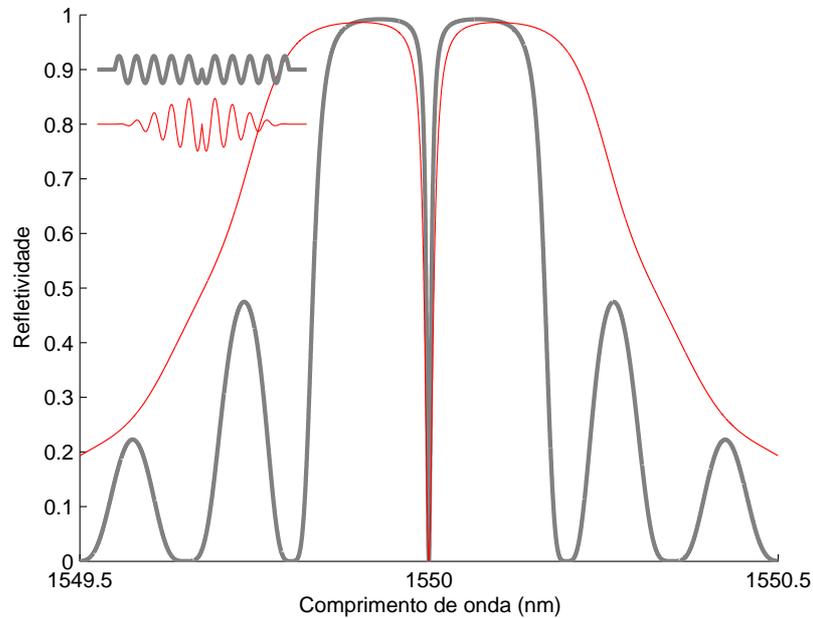


Figura 2.6: Espectros de uma grade uniforme e outra com apodização cosseno-levantado em ν , ambas com um deslocamento de fase de π a meio comprimento.

A Fig. 2.7 compara o mesmo espectro da grade uniforme com deslocamento de π , mostrado na Fig. 2.6, com outros dois espectros de duas grades similares, nas quais a única diferença em relação à grade original deve-se aos valores dos ângulos de deslocamento de fase: $\pi/2$ e $-\pi/2$. O efeito da modificação do ângulo é o deslocamento da fenda de transmissão no interior da banda de reflexão [21]. Esta característica pode ser bastante atraente para os sistemas WDM, servindo de base na construção de demultiplexadores ópticos [20]. A Fig. 2.8 exibe dois espectros de grades uniformes similares àquela cujo espectro é mostrado na Fig. 2.6, também com deslocamentos de π , porém localizados em diferentes posições da FBG: em $z = 2.5\text{mm}$ e $z = 4.0\text{mm}$ respectivamente. Quanto mais distante a posição de deslocamento de fase do centro da FBG, maior é a redução da profundidade da fenda [21].

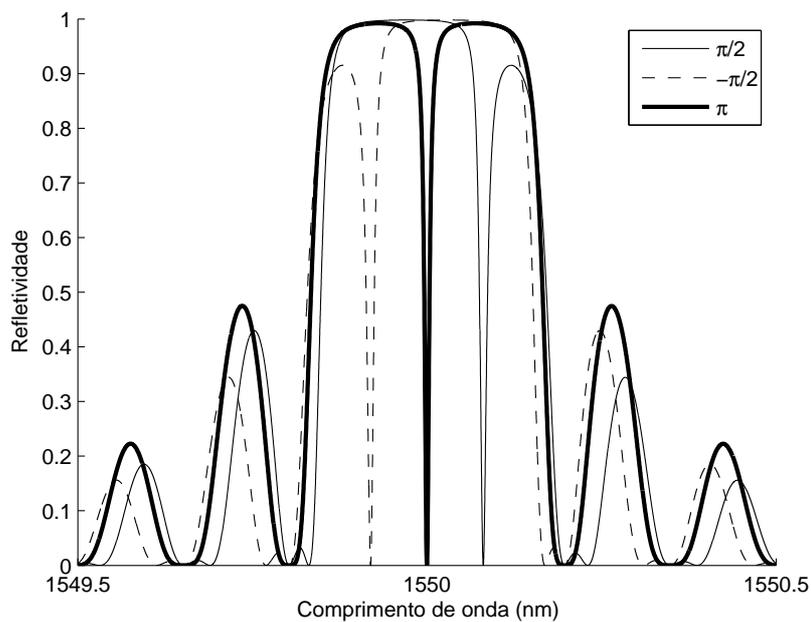


Figura 2.7: Efeito do ângulo de deslocamento sobre a posição da fenda no interior da banda de reflexão.

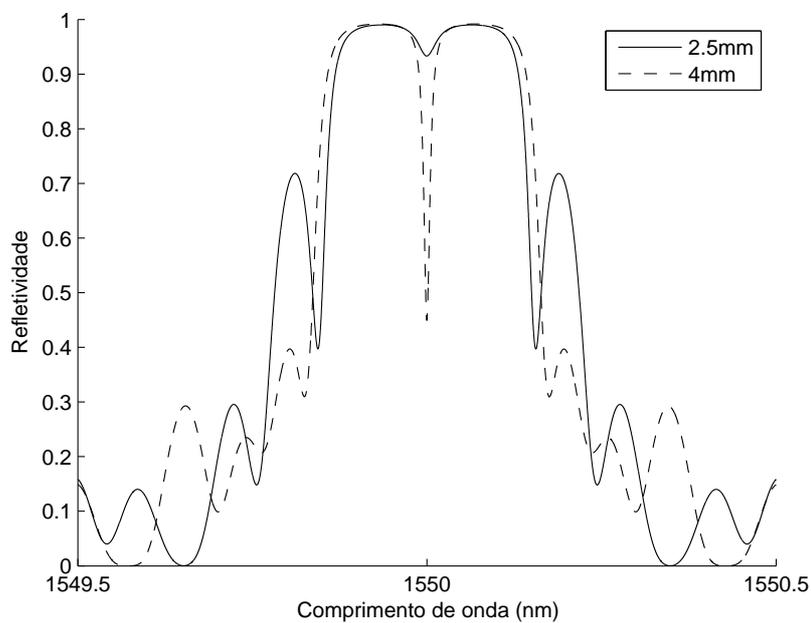


Figura 2.8: Efeito da posição do deslocamento sobre a profundidade da fenda no interior da banda de reflexão.

A Fig. 2.9 mostra dois espectros, (a) e (b), de duas grades uniformes dotadas respec-

tivamente de dois e três deslocamentos de fase de π . Os deslocamentos foram realizados em posições homoganeamente espaçadas ao longo das grades, em $z = L/3$ e $z = 2L/3$ para (a); $z = L/4$, $z = L/2$ e $z = 3L/4$ para (b). Esta figura demonstra a possibilidade de abrir mais de uma fenda do interior da banda de reflexão.

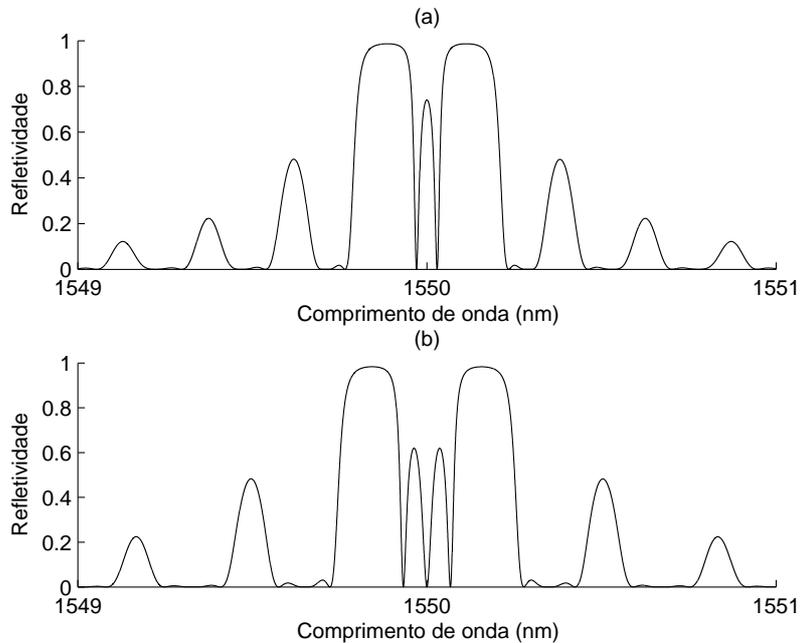


Figura 2.9: Efeito da ocorrência de mais de um deslocamento de π homoganeamente espaçados no interior da grade: dois deslocamentos em (a) e três em (b).

2.2.3 Grades com *chirp*

O *chirp* é a variação do período Λ em função da posição da grade z . Este tipo de perturbação de índice é geralmente realizado através das técnicas de fabricação utilizando escrita direta ou máscaras de fase. Para este propósito, as máscaras de fase são dotadas *chirp*, podendo possuir variações de periodicidade sofisticadas para além de uma variação linear, embora já tenha sido reportado o uso de máscaras de período uniforme na obtenção de grades com *chirp* através de um processo alternativo [23].

A análise de grades com *chirp* é automática através da formulação matricial apresentada, bastando a definição de valores diferentes para Λ_k . A Fig. 2.10 mostra os espectros de refletividade para três grades apodizadas cosseno-levantado em ν com $\delta n \rightarrow 0$ e máxima amplitude de perturbação $\delta n \times \nu = 4 \times 10^{-4}$. A primeira possui todas as seções com a mesmo período (*chirp* ausente) $\Lambda_k = 0.5\lambda_D/n_{eff}$, com $\lambda_D = 1550\text{nm}$ e $n_{eff} = 1.45$. A segunda grade possui um *chirp* linear de 0.1nm/mm a partir de $\Lambda_1 = 1549.5\text{nm}$ e a terceira possui um *chirp*

de 0.2nm/mm a partir de $\Lambda_1 = 1549\text{nm}$. O efeito do *chirp* mostrado é o alargamento do pico de máxima reflexão.

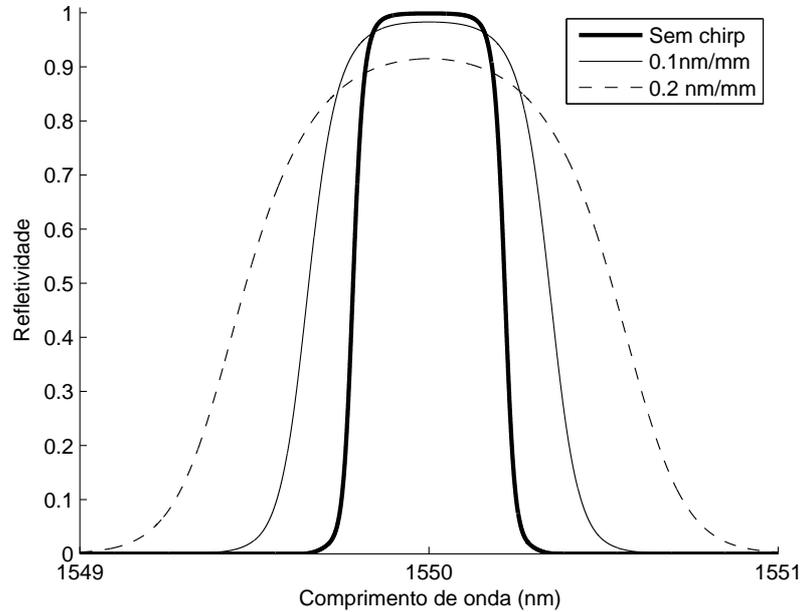


Figura 2.10: Efeito do *chirp* do alargamento na largura de banda de reflexão da FBG.

De certo modo o *chirp* associa cada posição da grade a um valor de Λ_D distinto, permitindo que diferentes comprimentos de onda reflitam em profundidades diferentes no interior da FBG, resultando em uma forte variação do atraso em função do comprimento de onda. O atraso A é calculado a partir da fase θ , $\theta = \angle\rho$, como [16]:

$$A = -\frac{\lambda^2}{2\pi c} \frac{d\theta}{d\lambda} \quad (2.15)$$

onde \angle representa o arco-tangente da parte imaginária do argumento dividido pela parte real; c representa a velocidade da luz no vácuo. A Fig. 2.11 exibe as curvas de atraso para as três grades apodizadas cujos espectros de refletividade são mostrados na Fig. 2.10. O formato das curvas de atraso para as grades com *chirp* apresentam uma inclinação monótona crescente, enquanto a grade sem *chirp* exibe uma curva de atraso simétrica e repleta de singularidades. As singularidades estão associadas às descontinuidades da curva de fase, cuja origem se deve principalmente ao comportamento numérico descontínuo do operador \angle , nas imediações dos locais onde a refletividade se anula (os nulos só podem ser observados através da curva de refletividade em dB: vide Fig. 2.4). Algumas operações de suavização e correção poderiam eliminar ou amenizar muitas destas irregularidades.

A dispersão D pode ser calculada a partir do atraso A como $D = dA/d\lambda$. É possível

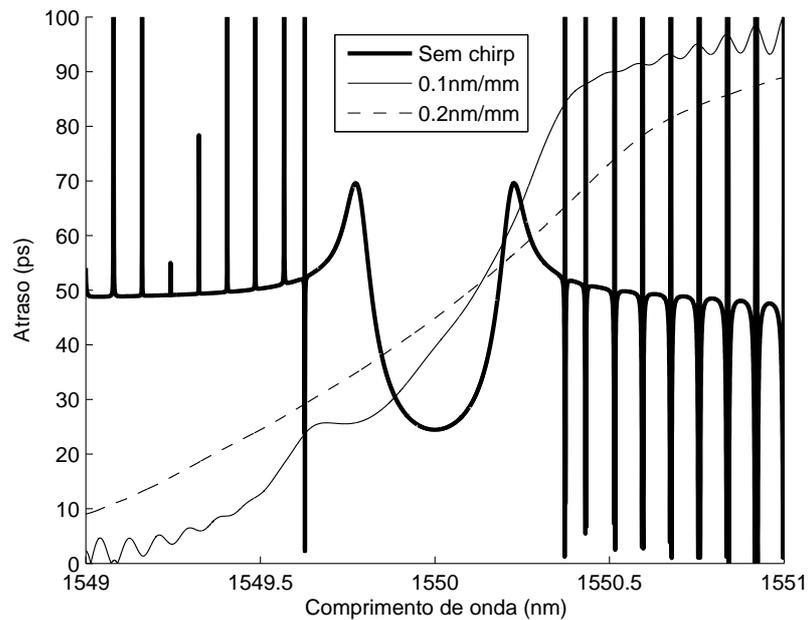


Figura 2.11: Efeito da posição do *chirp* do alargamento na largura de banda de reflexão da FBG.

projetar o *chirp* de uma grade de modo a obter elevados valores de dispersão negativa ou positiva, o que ratifica as FBG como importantes componentes para as redes ópticas também na compensação de dispersão. Outro benefício é a possibilidade de projetar filtros ópticos de alta refletividade que não afetem os pulsos que lhes atravessarem, portanto, apresentando dispersão nula.

Capítulo 3

Síntese de grades utilizando algoritmo genético e de enxame de partículas

O uso de heurísticas para projeto de filtros ópticos em fibra é algo bastante recorrente na literatura. A razão para isso se deve às dificuldades inerentes das técnicas inversas convencionais, como o algoritmo iterativo Gelfand–Levitan–Marchenko (GLM) [24] e as técnicas inversas baseadas no “descascamento” (*Layer Peeling* – LP) [3], incapazes de satisfazer simultaneamente a incontáveis restrições associadas às técnicas de fabricação. Por exemplo, os projetos que contemplam podem resultar repletos de deslocamentos de fase, demandando processos de fabricação como o da escrita ponto a ponto, cujo aparato é mais complexo e dispendioso. Por outro lado, heurísticas como o GA ou o PSO podem conseguir um equilíbrio muito interessante entre a qualidade e a complexidade dos projetos, permitindo satisfazer de forma flexível as restrições associadas aos processos de fabricação. Este capítulo explora estes dois algoritmos, introduzindo-os à síntese de grades.

3.1 GA

O Algoritmo Genético possui um histórico de quase vinte anos de extensa aplicação nas mais diversas áreas das ciências e das engenharias. Seu valor deve-se a sua flexibilidade e principalmente por ser uma heurística considerada, com certas reservas, global [5].

A concepção do GA é fruto dos trabalhos de John Holland e colaboradores da universidade de Michigan em meados da década 1970. Seus estudantes continuaram as pesquisas e atualmente, graças a teóricos como Goldberg [5] e Michalewicz [25], o GA é uma das mais documentadas heurísticas, com exemplos de aplicações bem sucedidas em quase todas as áreas

do conhecimento humano.

O GA, como o nome sugere, procura copiar os princípios naturais da genética. A heurística funciona através da simulação de uma população de soluções candidatas representadas por estruturas de dados cuja função é uma paródia dos cromossomos naturais dos quais emprestam o nome. As populações simuladas são sujeitas a operações de permuta de dados que procuram imitar os efeitos da reprodução. A evolução é obtida através da seleção, um mecanismo estocástico que é aplicado sobre a população simulada de modo a multiplicar a ocorrência das melhores soluções em detrimento das piores para a geração seguinte. A seleção, à sombra do que representa na natureza, é realizada considerando a aptidão das soluções candidatas. No GA, a aptidão pode ser representada por um valor ou por um conjunto de valores numéricos. Quanto maior a aptidão da solução, maior a probabilidade de ser selecionada e assim integrar a geração seguinte.

Os valores relacionados à aptidão são obtidos através da avaliação da solução candidata. Trata-se da averiguação do quão próxima a solução em questão encontra-se dos padrões estabelecidos. Para muitos problemas de engenharia, este processo é considerado crítico, pois pode recorrer à análise de estruturas ou sistemas complexos. A síntese de antenas é um exemplo interessante pois pode exigir métodos numéricos de análise específicos para obtenção, por exemplo, do espectro de refletividade de entrada ou do diagrama de radiação. Métodos dos Momentos, baseados em Diferenças Finitas ou Elementos Finitos podem ser necessários dependendo da complexidade das estruturas que precisem ser avaliadas; tais métodos são computacionalmente ostensivos, o que justifica a limitada aplicação (até o ano de 2005) de heurísticas para a síntese de antenas de geometrias mais complexas do que as filamentosas [26]. De fato, para cada nova solução candidata ou para qualquer solução modificada é necessário atualizar os parâmetros que definem sua aptidão. Conseqüentemente, a avaliação das soluções pode ser repetida milhares de vezes ao longo do processo de otimização ou síntese, multiplicando a demanda por competência e recursos computacionais.

-
01. Cria uma população \mathbf{A} aleatória;
 02. Para cada geração:
 03. $\mathbf{B} = \text{Muta\c{c}\~{a}o}(\text{Cruzamento}(\mathbf{A}))$;
 04. Para cada elemento k de \mathbf{B} :
 05. $\mathbf{B}[k].\text{aptid\~{a}o} = \text{FO}(\mathbf{B}[k])$;
 06. $\mathbf{A} = \text{Sele\c{c}\~{a}o}(\mathbf{B})$;
-

Figura 3.1: Algoritmo Genético básico

Alguns recursos das linguagens orientadas a objetos são empregados no pseudo-código

do GA básico mostrado na Fig. 3.1. Os objetos **A** e **B** representam populações formadas por vários indivíduos (cromossomos). As funções “Cruzamento” e “Mutaç o” realizam a reproduç o tendo como progenitores os indiv duos da populaç o **A** e resultando uma outra populaç o que   armazenada em **B**. A Funç o Objetivo   representada pelo operador “FO”, que realiza a avaliaç o do indiv duo argumento **B**[*k*] e retorna a aptid o correspondente, a qual   armazenada em um campo “*aptid o*” do pr prio objeto indiv duo. Finalmente, o operador “Seleç o” substitui a populaç o **A** utilizando novos indiv duos selecionados a partir da populaç o **B**.

3.1.1 GA bin rio

A literatura cl ssica, por J. H. Holland e Goldberg [5], gravita fundamentalmente em torno dos GA bin rios. De fato, os princ pios b sicos de todo GA podem ser melhor compreendidos a partir da vers o bin ria, no qual as soluç es candidatas s o representadas por “cromossomos”, constitu dos por seq ncias de genes ou *bits*. Considere o exemplo simples de minimizaç o da funç o de Rastrigin bidimensional [27]:

$$f(x_1, x_2) = x_1^2 + x_2^2 - 10[2 - \cos(2\pi x_1) - \cos(2\pi x_2)], \quad (3.1)$$

que possui valor m nimo conhecido $f(x_1, x_2) = 0$ para $x_1 = x_2 = 0$. Considere o espaço de soluç es definido por $|x_k| < 2$ com $k = 1, 2$. Uma soluç o candidata pode ser representada nesse espaço atrav s de uma seq ncia de 16 bits **S**, por exemplo **S** = {0011101110011011}. Convenciona-se que os 8 primeiros bits de **S** sejam referentes a x_1 enquanto os 8 bits finais a x_2 , ou seja, **S** = { s_1, s_2 } onde $s_1 = \{00111011\}$ e $s_2 = \{10011011\}$. Os valores de x_k podem ser obtidos a partir de s_k atrav s de:

$$x_k = x_k^{min} + (x_k^{max} - x_k^{min}) \frac{int(s_k)}{2^{N_k} - 1} \quad (3.2)$$

onde x_k^{min} e x_k^{max} representam os extremos do espaço de soluç es para a dimens o k ; *int* representa uma funç o capaz de decodificar uma seq ncia bin ria em um n mero inteiro e N_k representa o n mero de bits de s_k . Para o exemplo em quest o, x_k^{min} e x_k^{max} s o respectivamente -2 e 2 para $k = 1, 2$. A funç o *int* retorna respectivamente 59 e 155 para s_1 e s_2 , logo $x_1 = -1.0745$ e $x_2 = 0.43137$. O espaço de soluç es   constitu do por 65536 pontos, resoluç o associada ao erro absoluto m ximo de aproximadamente 0.015686.

A Fig. 3.2 mostra a funç o $f(x_1, x_2)$ para o intervalo $|x_k| < 2$. O aspecto multimodal da funç o inviabiliza o uso de t cnicas cl ssicas de busca.

O modo como as soluções são codificadas na forma de um cromossomo \mathbf{S} para este exemplo pode ser apropriado para inúmeros problemas, para tanto \mathbf{S} pode ser generalizado como $\mathbf{S} = \{s_1, s_2, \dots, s_{k-1}, s_k, s_{k+1}, \dots, s_{N_D-1}, s_M\}$, onde N_D é o número de dimensões (ou de parâmetros) do problema. Cada elemento de \mathbf{S} , s_k , pode possuir um número de bits N_k diferente.

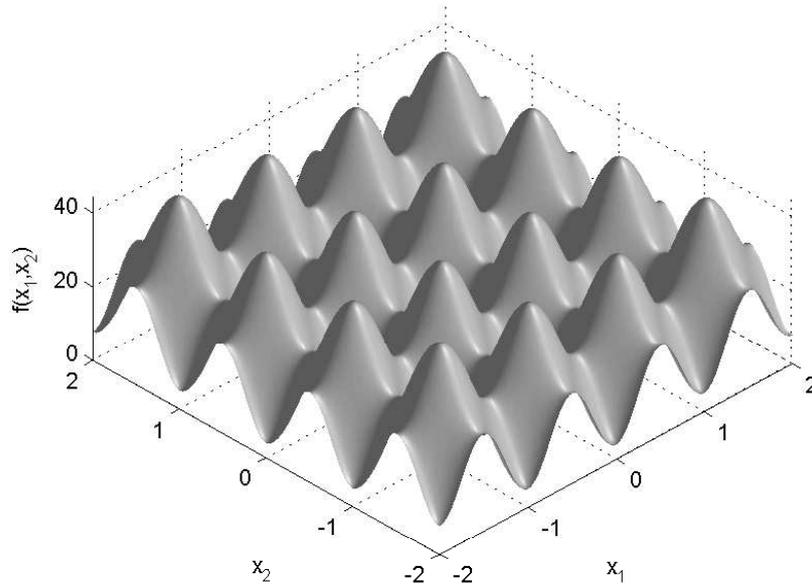


Figura 3.2: Função Rastrigin bidimensional para $|x_k| < 2$.

3.1.1.1 Operador de cruzamento

O operador de cruzamento faz parte do mecanismo de reprodução do GA utilizado para criar novas soluções a partir de outras existentes. Trata-se de um procedimento estocástico de troca de genes que procura imitar a reprodução, na qual os materiais genéticos dos progenitores são mesclados para formar um novo indivíduo. A origem do termo provém da biologia, para descrever o emparelhamento, ruptura e consolidação de dois cromossomos naturais correspondentes durante a meiose celular. Este comportamento é simulado no cruzamento entre dois cromossomos artificiais: partes de ambos são comutadas tal como mostrado na Fig. 3.3.

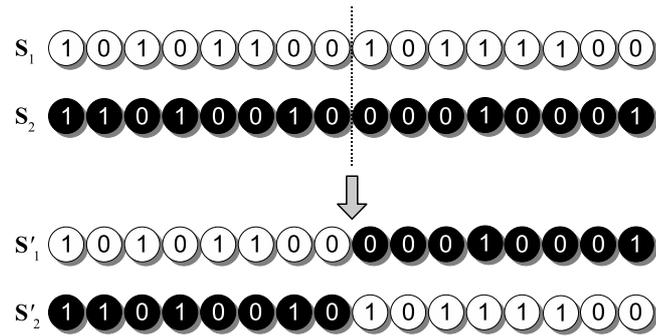


Figura 3.3: Cruzamento de ponto único.

A partir dos cromossomos progenitores S_1 e S_2 resultam os filhos S'_1 e S'_2 . Cada filho é constituído por duas partes contínuas oriundas de cada progenitor, como mostrado na 3.3. Este conceito pode ser estendido, originando o cruzamento de dois pontos de ruptura como mostrado na Fig. 3.4. Um número arbitrário de pontos ruptura, porém inferior ao comprimento de S_1 e S_2 , podem ser utilizados, generalizando operador cruzamento (cruzamento multi-ponto). O caso extremo resulta no cruzamento uniforme, onde os genes são comutados em função de uma probabilidade, normalmente de 50% tal como mostrado na Fig. 3.5.

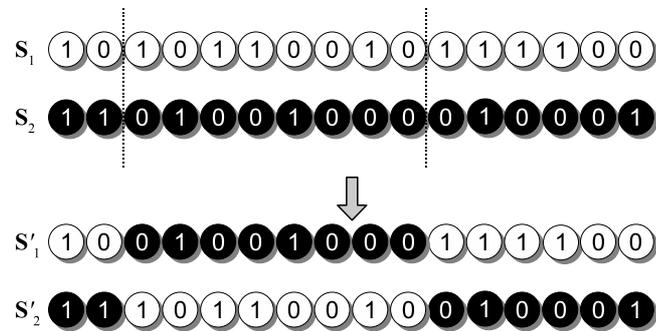


Figura 3.4: Cruzamento de dois pontos.

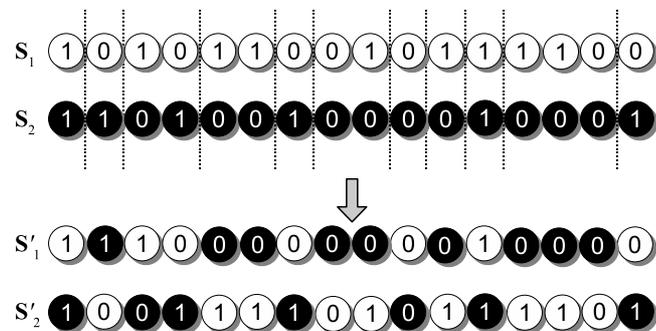


Figura 3.5: Cruzamento uniforme.

Se o operador cruzamento for aplicado a todos os indivíduos de uma geração, alguns cromossomos promissores podem ser modificados e perdidos. Portanto, é interessante associar uma probabilidade P_C ao operador cruzamento. Portanto, a quantidade percentual de cromossomos resultantes de cruzamento na população retornada pelo operador será de aproximadamente $100 \times P_C\%$, enquanto a quantidade de cromossomos inalterados será de $100 \times (1 - P_C)\%$. De acordo com observações empíricas, aconselha-se utilizar $P_C > 0.5$ [5].

O operador de cruzamento é capaz de criar novos indivíduos, que exploram novos locais dentro do espaço de soluções do problema provavelmente distantes das posições associadas aos cromossomos progenitores. Evidentemente esses novos indivíduos podem ser qualitativamente inferiores aos pais, mas caso sejam, estes serão removidos ao longo das gerações pela seleção; novas e melhores soluções persistirão no lugar das soluções inferiores garantindo o progresso do processo de busca. Mas o fato dos cromossomos filhos serem diferentes dos pais favorece a manutenção da diversidade das populações, isto é, o quão diferentes os indivíduos são uns dos outros. A diversidade pode impedir a convergência prematura a um ótimo local, conservando o GA continuamente engajado na busca pelo ótimo global.

3.1.1.2 Operador de mutação

A mutação é outro operador importante que permite explorar o espaço de soluções gerando soluções alternativas que podem ser superiores ou inferiores em relação aos originais. Porém a mutação funciona de uma forma bastante distinta do cruzamento. Dentro do GA binário, a mutação é um processo estocástico que inverte arbitrariamente alguns bits dos cromossomos em função de uma probabilidade P_M . De acordo com [5], a mutação deve modificar sutilmente os cromossomos de modo a proporcionar uma exploração de locais do espaço de soluções próximos dos cromossomos originais. Também pode, assim como o cruzamento, combater a perda de diversidade da população, impedindo a convergência prematura e conservando o GA continuamente engajado na busca pelo ótimo global.

O operador “Mutação” do algoritmo da Fig. 3.1 opera sobre toda a população retornada pelo operador “Cruzamento”, invertendo cada gene de cada cromossomo com uma probabilidade de P_M (Fig. 3.6). Este comportamento tenta imitar o correspondente fenômeno natural que tende a ocorrer de forma espontânea durante o processo de meiose celular. Entretanto, o percentual de alterações dos cromossomos naturais devido a mutação é insignificante. De fato, P_M costuma assumir valores realmente pequenos, quando comparados com P_C , usualmente entre 0.001 e 0.1 [5].

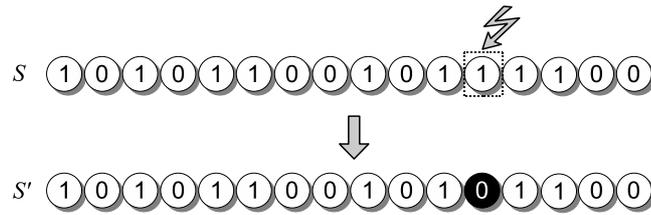


Figura 3.6: Mutação.

3.1.2 GA real

A codificação binária possui desvantagens. Por exemplo, dois cromossomos podem se diferir por apenas 1 bit, porém representar dois números reais completamente diferentes e distantes. Este problema é conhecido como problema de *Hamming cliffs* e pode prejudicar os algoritmos genéticos binários [5]. Quando o operador mutação inverte arbitrariamente apenas um bit de um cromossomos, corre o risco de fazê-lo com algum bit de elevada ordem. O resultado pode ser um deslocamento exagerado da solução candidata no interior do espaço de buscas: um comportamento incompatível com as prerrogativas do operador mutação. Para combater este problema, normalmente recorre-se à codificação de Gray. O código Gray, também conhecido como “código binário refletido”, é um sistema numérico em que valores contíguos se diferenciam por apenas um bit. Isso significa que o impacto da modificação de um bit, mesmo que de mais elevada ordem, será menor do que para o caso utilizando o sistema binário convencional. A implementação do código refletido pode ser feita sobre da função $int()$ de (3.2), cujo algoritmo é mostrado na Fig. 3.7.

O uso do código Gray permite realizar uma busca gradual, aplacando o problema *Hamming cliffs*. Sua superioridade em relação ao uso do sistema binário natural tem sido comprovada para a maioria dos problemas [28]. Por outro lado, independente do sistema binário, o espaço de buscas será sempre discretizado, não sendo possível soluções com precisão arbitrária. Um outro aspecto negativo da codificação binária resulta da relação de proporcionalidade entre o comprimento dos cromossomos e o tamanho das populações [5]. Grandes precisões numéricas são conseguidas a custo de cromossomos longos que, conseqüentemente, demandam o uso de grandes populações. Como o GA emprega um número massivo de chamadas à FO, proporcional ao tamanho da população, a demanda computacional pode aumentar a um ponto intolerável.

Por outro lado, a codificação real emprega um valor para cada parâmetro do problema. Para o problema de minimização da função Rastrigin bidimensional (Fig. 3.2), dois parâmetros seriam codificados no interior de cada cromossomo de 16 bits, cada um deles

Parâmetros:

s_k : o número binário codificado

N_k : comprimento de s_k

Retorna:

x : número inteiro decimal

01. $aux1 = 2^{N_k-1}$;

02. $aux2 = s_k[N_k]$;

03. $x = aux2 \times aux1$;

04. Para $j = -1$ até 1:

05. $aux2 = aux2 \text{ xor } s_k[j]$;

06. $aux1 = aux1/2$;

07. $x = x + aux2 \times aux1$;

08. retorna x ;

Figura 3.7: Função *int* para a decodificação de um número binário refletido.

utilizando 8 bits, permitindo uma precisão bastante limitada. Entretanto, um cromossomo utilizando codificação real seria $\mathbf{S} = \{x_1, x_2\}$, empregando diretamente as coordenadas do espaço euclidiano como genes ponto-flutuante, o que permitiria uma precisão praticamente arbitrária. A redução da dimensionalidade do problema certamente melhoraria a convergência do GA, além de reduzir a demanda por grandes populações [28]. Uma generalização dos cromossomos reais seria algo tão simples como $\mathbf{S} = \{x_1, x_2, \dots, x_{k-1}, x_k, x_{k+1}, \dots, x_{N_D-1}, x_{N_D}\}$, isto é, um vetor de comprimento N_D constituído pelas próprias variáveis do problema. Esta forma de representação simples e direta pode ser bastante interessante dada a existência de bibliotecas computacionais para cálculo vetoriais e até mesmo de computadores com arquiteturas vetoriais.

3.1.2.1 Operador de cruzamento

A concepção de operadores genéticos para o GA com codificação real tem sido um desafio. Incontáveis tipos de cruzamento têm sido propostos na literatura, donde destacam-se o cruzamento discreto multiponto, o cruzamento intermediário, o BLX e o SBX [29].

O cruzamento discreto multiponto imita o cruzamento utilizado com a codificação binária realizado através da comutação de partes dos cromossomos progenitores; a única diferença em relação ao cruzamento de múltiplos pontos de ruptura feitas sobre os cromossomos binários está na natureza dos genes, que são pontos-flutuantes. No caso mínimo, apenas

1 ponto de cruzamento é empregado. O caso extremo resulta em algo similar ao cruzamento uniforme, em que todos genes dos cromossomos pais possuem uma certa probabilidade de serem comutados.

O cruzamento intermediário utiliza uma combinação linear entre os elementos dos cromossomos progenitores para criar os elementos dos cromossomos filhos. Considere os cromossomos progenitores como \mathbf{S}_1 e \mathbf{S}_2 . Os cromossomos filhos \mathbf{S}'_1 e \mathbf{S}'_2 podem ser obtidos respectivamente por

$$\mathbf{S}'_1[k] = \mathbf{S}_1[k]u_k + \mathbf{S}_2[k](1 - u_k) \quad (3.3)$$

e

$$\mathbf{S}'_2[k] = \mathbf{S}_1[k](1 - u_k) + \mathbf{S}_2[k]u_k, \quad (3.4)$$

onde o subscrito k indica o número do gene e $u_k = u(0, 1)$, sendo $u(0, 1)$ a realização de uma variável aleatória uniforme entre 0 e 1.

O cruzamento BLX (*Blend Crossover*) é bastante semelhante ao cruzamento uniforme. Os genes dos filhos também são obtidos através de uma combinação linear dos genes paternos, porém a variável u_k passa a ser definida como $u_k = (1 + 2\alpha)u(0, 1) - \alpha$. Na prática este tipo de cruzamento é um caso mais geral do cruzamento uniforme que ocorre para $\alpha = 0$. Estudos experimentais demonstraram que o valor mais adequado de α para muitos problemas simples parece ser 0.5, circunstância em que o valor mínimo e máximo de p_k igualam-se a -0.5 e 1.5 respectivamente. Isso implica que os genes dos filhos não serão mais restringidos para o intervalo definido pelos genes paternos, dando ao GA um maior poder de exploração do espaço de buscas.

O SBX (*Simulated Binary Crossover*) simula os efeitos do cruzamento de cromossomos binários utilizando apenas um ponto de ruptura. Até certo ponto, o procedimento é muito parecido com o BLX e também com o cruzamento intermediário: os filhos também são obtidos através de uma combinação linear entre os genes dos pais. Porém, diferentemente destes outros tipos de cruzamento, o processo é parametrizado por variáveis aleatórias não uniformes. Os cromossomos filhos \mathbf{S}_1 e \mathbf{S}_2 podem ser definidos em função dos pais \mathbf{S}_1 e \mathbf{S}_2 :

$$\mathbf{S}'_1[k] = \frac{1}{2} [(1 + u'_k)\mathbf{S}_1[k] + (1 - u'_k)\mathbf{S}_2[k]] \quad (3.5)$$

$$\mathbf{S}'_2[k] = \frac{1}{2} [(1 - u'_k)\mathbf{S}_1[k] + (1 + u'_k)\mathbf{S}_2[k]] \quad (3.6)$$

Onde $k = 1, 2, 3, \dots, N_D$ e u_k é definido como:

$$u'_k = \begin{cases} 2u_k^{\frac{1}{\eta+1}} & \text{se } u_k \leq \frac{1}{2} \\ 2(1-u_k)^{-\frac{1}{\eta+1}} & \text{se } u_k > \frac{1}{2} \end{cases} \quad (3.7)$$

Onde $u_k = u(0, 1)$. O parâmetro η controla a variância do processo, permitindo uma maior probabilidade dos indivíduos filhos resultarem diferentes dos pais quanto menor for seu valor. η precisa ser cuidadosamente definido em função do tipo de problema.

Investigações empíricas revelaram que, para um dado problema, para cada valor de α existirá um η correspondente, os quais proporcionarão aos algoritmos genéticos utilizando respectivamente cruzamentos BLX e SBX desempenhos similares. Para BLX, um valor típico para α é de aproximadamente 0.5, para o qual corresponde o valor $\eta = 3$ para o SBX [29].

Os procedimentos de cruzamento sobre cromossomos reais, tal como para os cromossomos binários, são normalmente associados a uma probabilidade P_C .

3.1.2.2 Operador de mutação

A mutação se manifesta através de pequenas mudanças nos cromossomos, possibilitando a exploração do espaço de soluções nas proximidades do cromossomo original. Para obter tal efeito em cromossomos binários, a mutação binária comuta em média apenas 1 bit aleatório. Espera-se, com isso, uma pequena modificação do indivíduo, a despeito do problema de *Hamming cliffs*. Para conseguir um processo similar com cromossomos constituídos por números reais, os procedimentos de mutação apelam para desvios que seguem distribuições probabilísticas variadas. Basicamente são utilizadas as distribuições uniforme e gaussiana [25].

Na mutação uniforme, o cromossomo após a mutação \mathbf{S}' é obtido a partir do cromossomo original \mathbf{S} através da seguinte relação:

$$\mathbf{S}'[k] = \mathbf{S}[k] + (u(0, 1) - 0.5)\sigma_{Mk}, \quad (3.8)$$

onde σ_{Mk} determina o máximo desvio permitido para o gene $\mathbf{S}[k]$.

A mutação gaussiana utiliza a realização uma distribuição gaussiana de média nula e desvio padrão 1 representada por $n(0, 1)$:

$$\mathbf{S}'[k] = \mathbf{S}[k] + \sigma_{Mk}n(0, 1), \quad (3.9)$$

onde σ_{Mk} representa, neste caso, o desvio padrão associado ao gene k .

3.1.2.3 Observação das restrições

Diferentemente da codificação binária que necessita definir as variáveis do problema em intervalos para proceder a discretização, o GA real permite que os operadores genéticos explorem livremente as variáveis ignorando quaisquer limites. Para alguns problemas essa característica pode ser fundamental, para outros essa capacidade é absolutamente desnecessária ou até indesejável. Os problemas de engenharia geralmente conhecem muito bem suas restrições, pois são sempre associadas à viabilização matemática dos modelos, viabilização dos processos de manufatura ou mesmo associadas à viabilização econômica dos projetos.

De acordo com as subseções anteriores, os operadores de cruzamento como o BLX e o SBX, além dos operadores de mutação, são capazes de alterar os parâmetros de uma solução candidata sem temer que o resultado extrapole algum intervalo. A forma mais simples de se corrigir o problema seria mover x_k novamente para o intervalo admissível $[x_k^{min}, x_k^{max}]$, por exemplo, através da seguinte condição:

$$x_k = \begin{cases} x_k^{min} & se & x_k < x_k^{min} \\ x_k^{max} & se & x_k > x_k^{max} \\ x_k & se & x_k^{min} \leq x_k \leq x_k^{max} \end{cases} \quad (3.10)$$

A Fig. 3.8 exhibe as distribuições para x_k originalmente igual a zero após um operador de mutação gaussiano com desvio padrão igual a 1. A curva tracejada é a distribuição após a aplicação do operador sem a aplicação de qualquer tipo de restrição. A curva cheia é a distribuição truncada para o intervalo $[x_k^{min}, x_k^{max}] = [-2, 2]$. Nota-se facilmente que os valores extremos do intervalo assumem probabilidades muito elevadas. Este aspecto é prejudicial ao GA, pois poucos valores específicos podem se tornar muito frequentes para as variáveis do problema, comprometendo a diversidade e a eficácia da heurística.

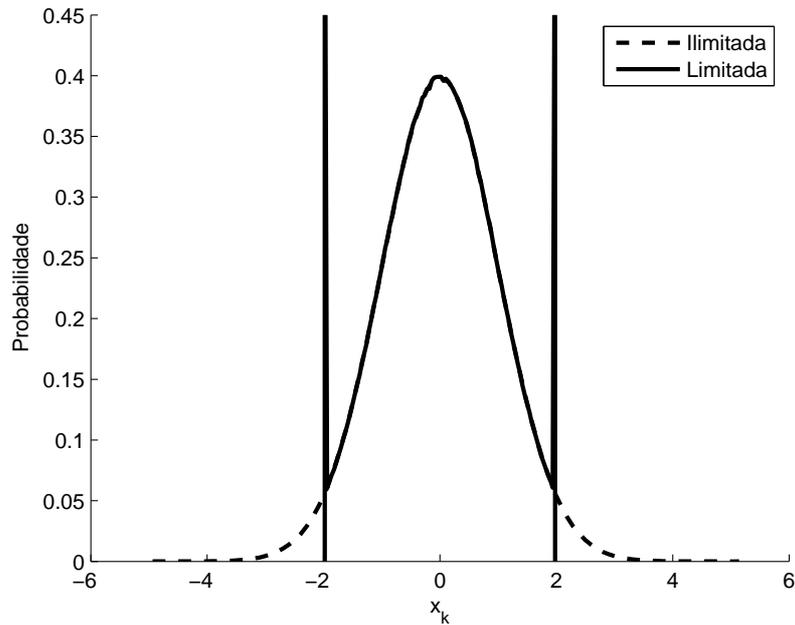


Figura 3.8: Densidades para uma variável submetida a um desvio gaussiano sem e com truncagem simples.

Para evitar o problema da truncagem simples, pode-se recorrer a re-execução do operador genético para o gene x_k enquanto este resultar fora do intervalo $[x_k^{min}, x_k^{max}]$. Tal como mostrado na Fig. 3.9, o efeito deste processo é uma distribuição cortada nos limites, porém sem singularidades prejudiciais como as mostradas na Fig. 3.8. Para distribuições de variância modesta, existe uma grande probabilidade de que novas “tentativas” do operador genético não violem as restrições impostas.

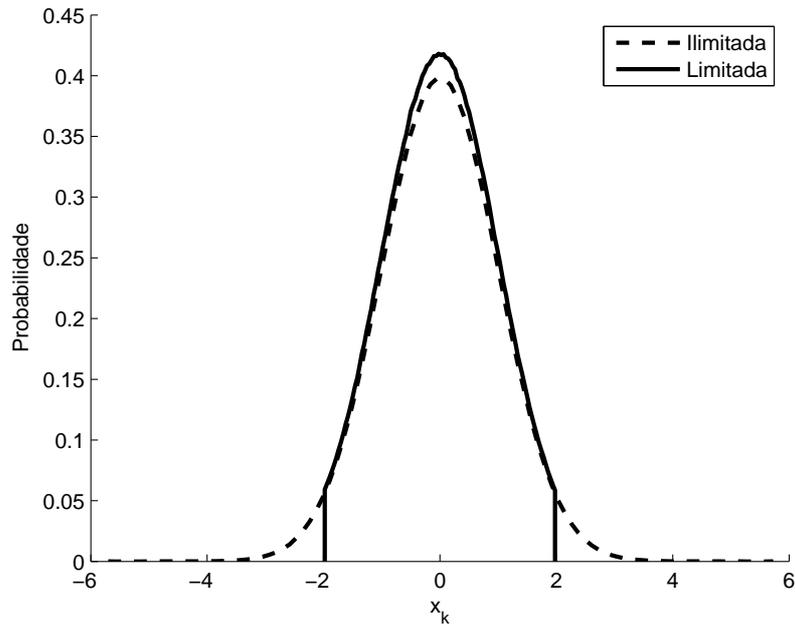


Figura 3.9: Densidades para uma variável submetida a um desvio gaussiano sem e com truncagem utilizando re-execução.

Para operadores cujas distribuições possuam varianças da ordem da extensão do próprio intervalo, a estratégia de re-execução do operador pode aumentar exageradamente o custo computacional do processo devido a ocorrência de muitas tentativas frustradas. Neste caso é possível adotar uma abordagem alternativa: caso a componente desviada x_k resulte fora do intervalo $[x_k^{min}, x_k^{max}]$, uma distribuição uniforme é adotada de modo que $x_k = U(x_k^{min}, x_k^{max})$, isto é, uma distribuição uniforme entre x_k^{min} e x_k^{max} . Como mostrado na Fig. 3.10, o efeito da sobreposição de distribuição uniforme é bem próximo do conseguido com a estratégia da re-execução.

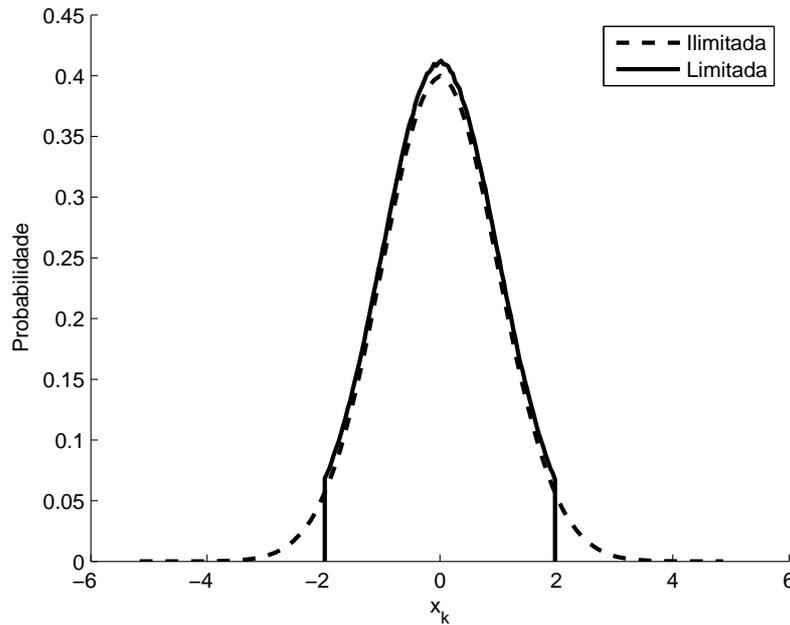


Figura 3.10: Densidades para uma variável submetida a um desvio gaussiano sem e com truncagem utilizando a sobreposição de distribuição uniforme.

O uso de estratégias ou regras para a obediência das restrições do problema torna desnecessário o uso de um algoritmos com penalidades, o que, de uma forma geral, simplifica a implementação da heurística. Portanto, este trabalho adotará sempre esta abordagem, utilizando sempre a estratégia da sobreposição de distribuição uniforme já embutida nos operadores genéticos.

3.1.3 Operador de seleção

Os operadores de cruzamento e de mutação não carregam em si quaisquer recursos para garantir a convergência do processo de busca. A seleção é tudo o que o GA possui dedicado a esta tarefa. A seleção é um procedimento estocástico que tem o objetivo de multiplicar a ocorrência de cromossomos promissores, isto é, mais aptos, em detrimento dos menos aptos. De forma semelhante como ocorre na natureza, a seleção geralmente tolera falhas, permitindo que soluções menos promissoras possam ter uma pequena chance de lograr êxito e passar à geração seguinte.

Uma vez que a aptidão associada a cada indivíduo é representada por um ou mais valores reais independentemente do tipo de codificação utilizada para os cromossomos, tanto o GA binário quanto o GA real compartilham dos mesmos tipos de operadores de seleção.

3.1.3.1 Seleção por roleta

Existem muitos métodos de seleção disponíveis na literatura. O Roulette–Wheel (RW) em essência é sem dúvidas um dos métodos mais simples. O RW deriva seu nome do seu princípio: similar ao funcionamento de uma roleta. Figurativamente, cada cromossomo da população \mathbf{B} (vide Fig. 3.1) ocuparia um certo setor de uma roleta com um comprimento de arco proporcional à sua aptidão.

Parâmetros:

\mathbf{B} : população para ser selecionada

Retorna:

\mathbf{A} : população selecionada

01. Para cada elemento i de \mathbf{A} :
02. $total = 0$;
03. Para cada elemento k de \mathbf{B} :
04. $total = total + \mathbf{B}[k].aptidão$;
05. Para cada elemento k de \mathbf{B} :
06. $\mathbf{V}[k] = \mathbf{B}[k].aptidão / total$;
07. $r = u(0, 1)$;
08. $k = 1$;
09. Enquanto $r > \mathbf{V}[k]$ e $k \leq \mathbf{B}.tamanho$:
10. $r = r - \mathbf{V}[k]$;
11. $k = k + 1$;
12. $\mathbf{A}[i] = \mathbf{B}[k]$;
13. retorna \mathbf{A} ;

Figura 3.11: Algoritmo para a seleção Roulette–Wheel

Como mostrado no algoritmo da Fig. 3.11, a roleta é representada pelo vetor \mathbf{V} cujo comprimento é igual ao tamanho da população \mathbf{B} . Cada valor de aptidão é normalizado em relação à somatória $total$ e armazenado em uma posição de V . Uma amostra aleatória r é capturada a partir de uma distribuição uniforme entre 0 e 1. A simulação do giro da roleta é realizada procurando a que intervalo r corresponde. Ajuda muito na compreensão do processo a Fig. 3.12, extraída de [29], onde o vetor V pode ser comparado a uma “roleta desenrolada”.

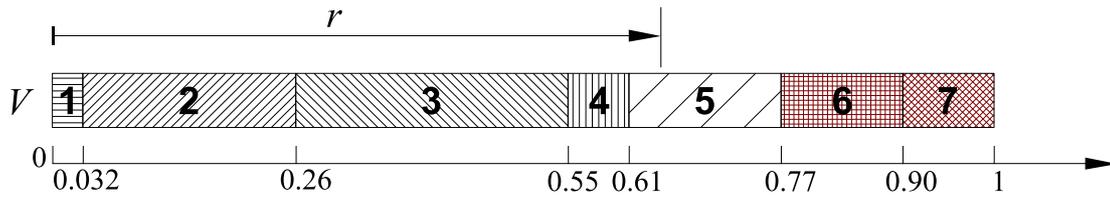


Figura 3.12: Implementação da roleta desenrolada.

3.1.3.2 Seleção por roleta com *ranking*

A seleção RW possui muitas limitações. Principalmente, porque depende diretamente dos valores de aptidão associados a cada cromossomo. Se algumas poucas soluções destacam-se demasiadamente dentre as demais da mesma população, o processo de seleção pode acabar ignorando muitos indivíduos promissores por ainda possuírem valores de aptidão medíocres. Neste caso, a diversidade da população da próxima geração será prejudicada. O uso da técnica chamada *ranking* elimina este problema: os valores de \mathbf{V} não são mais obtidos diretamente a partir dos valores de aptidão das soluções, mas da posição que ocupam dentro da população ordenada em função da aptidão. A Fig. 3.13 exhibe o algoritmo do processo de seleção RW com *ranking*, que tem como argumento a população \mathbf{B} e retorna a população \mathbf{A} . A ordenação de \mathbf{B} , em função da aptidão, é realizada na linha 01 pelo procedimento “Ordene”. De acordo com a linha 05, para este exemplo $\mathbf{V}[k]$ varia linearmente em função de k e sob o controle dos parâmetros a e b (*ranking* linear).

Os parâmetros a e b têm um papel muito importante no *ranking* linear, pois oferecem a possibilidade de controle da pressão seletiva. A pressão seletiva é o quanto o processo de seleção favorece os cromossomos mais aptos em detrimento dos menos aptos. Obviamente o RW simples é incapaz de oferecer tal controle. Porém uma pressão seletiva demasiadamente elevada pode levar à convergência prematura, enquanto baixas pressões podem tornar a convergência muito mais demorada do que o necessário. O *ranking* linear estabelece uma pressão seletiva proporcional ao módulo da diferença entre a e b .

3.1.3.3 Seleção por torneio

Na seleção por torneio, um subconjunto é formado a partir da população \mathbf{B} com indivíduos aleatoriamente sorteados. Deste subconjunto, o melhor indivíduo é escolhido para fazer parte da população \mathbf{A} . O processo é repetido até que a população \mathbf{A} seja satisfatoriamente preenchida.

Parâmetros:

\mathbf{B} : população para ser selecionada

a, b : argumentos escalares para o controle do *ranking* linear

Retorna:

\mathbf{A} : população selecionada

01. Ordene(\mathbf{B});
 02. Para cada elemento j de \mathbf{A} :
 03. $total = 0$;
 04. Para cada elemento k de \mathbf{B} :
 05. $\mathbf{V}[k] = a + (b - a) \times (k - 1) / (N - 1)$;
 06. $total = total + \mathbf{V}[k]$;
 07. Para cada elemento k de \mathbf{B} :
 08. $\mathbf{V}[k] = \mathbf{V}[k] / total$;
 09. $r = u(0, 1)$;
 10. $k = 1$;
 11. Enquanto $r > \mathbf{V}[k]$:
 12. $r = r - \mathbf{V}[k]$;
 13. $k = k + 1$;
 14. $\mathbf{A}[j] = \mathbf{B}[k]$;
 15. retorna \mathbf{A} ;
-

Figura 3.13: Seleção Roulette–Wheel com *ranking* linear

O tamanho do subconjunto de competição define a pressão seletiva do processo: quanto maior a capacidade do subconjunto, maior a pressão seletiva. Dentre as vantagens deste método destacam-se: (1) simplicidade de implementação, associada também a um menor custo computacional; (2) assim como a seleção RW com *ranking*, o processo não depende diretamente dos valores de aptidão associados aos cromossomos; (3) pode ser facilmente adaptado para problemas de maximização, de minimização, ou multiobjetivo; (4) não exige a coleta de dados de toda a população, portanto é mais simples de implementar em arquiteturas paralelas ou distribuídas [5].

Uma importante variante da seleção por torneio é a seleção por torneio estocástico, em que uma probabilidade de sucesso P_S é associada ao processo. Assim, mesmo o pior indivíduo do subconjunto passa a possuir uma certa probabilidade de ser selecionado. O parâmetro P_S atua também como mais um parâmetro de controle da pressão seletiva, capaz de realizar uma

espécie de “ajuste fino”. Mesmo que o tamanho de subconjunto seja mínimo (dois), a pressão seletiva pode ser reduzida ainda mais com a redução da probabilidade de sucesso. A seleção por torneio estocástico é esboçada no algoritmo da Fig. 3.14. Neste algoritmo o subconjunto de competição possui uma profundidade de N_T , sendo este parâmetro usado na definição da variável P_{N_T} . Na linha 03, b é iniciado com um valor inteiro pertencente ao intervalo $[0, \mathbf{B.tamanho})$, sendo *round* um operador de arredondamento para o menor mais próximo do argumento. O algoritmo percorre a população \mathbf{B} testando cada indivíduo $\mathbf{B}[k]$ com uma probabilidade P_{N_T} , de modo que, ao cabo do laço de k , aproximadamente N_T indivíduos de \mathbf{B} tenham sido testados. Na linha 07 o melhor dentre aproximadamente N_T indivíduos de \mathbf{B} é selecionado. O processo repete-se para cada posição da população \mathbf{A} .

Parâmetros:

\mathbf{B} : população para ser selecionada

N_T : capacidade do subconjunto do torneio

P_S : probabilidade de sucesso

Retorna:

\mathbf{A} : população selecionada

01. $P_{N_T} = N_T / \mathbf{B.tamanho}$; 02. Para cada elemento i de \mathbf{A} :

03. $b = \text{round}(u(1, \mathbf{B.tamanho}))$;

03. Se $u(0, 1) < P_S$:

04. Para cada elemento k de \mathbf{B} :

05. Se $u(0, 1) < P_{N_T}$ e $b \neq k$ e $\mathbf{B}[k]$ melhor do que $\mathbf{B}[b]$:

06. $b = k$;

07. $\mathbf{A}[i] = \mathbf{B}[b]$;

08. retorna \mathbf{A} ;

Figura 3.14: Seleção por torneio estocástico

Obviamente, fazendo $P_S = 1$ o algoritmo da Fig. 3.14 passaria a descrever a seleção por torneio determinístico.

3.1.3.4 Elitismo e a seleção elitista

O elitismo é realizado copiando um certo número dos melhores indivíduos da geração atual para a geração seguinte automaticamente. Para uma população constante de N_P indivíduos, N_E indivíduos da geração seguinte serão cópias dos melhores da geração anterior,

enquanto os outros $N_P - N_E$ serão determinados através da aplicação dos operadores genéticos e de seleção. A Fig. 3.15 mostra um algoritmo genético elitista. As linhas de 04 a 09 ordenam a população \mathbf{A} em função da aptidão, colocando os melhores N_E indivíduos nas primeiras posições. Na linha 10 a população \mathbf{A} é redimensionada conservando os primeiros N_E indivíduos. Finalmente, na linha 13, a população da geração seguinte é formada incorporando o resultado do procedimento de seleção sobre a população temporária \mathbf{B} (processo representado pela operação de “adição”). É importante observar que para $N_E = 0$ o algoritmo da Fig. 3.15 coincidirá com o primeiro algoritmo da Fig. 3.1.

```

01. Cria uma população  $\mathbf{A}$  aleatória;
02. Para cada geração:
03.      $\mathbf{B} = \text{Mutação}(\text{Cruzamento}(\mathbf{A}))$ ;
04.     Para  $i = 1$  e enquanto  $i \leq N_E$ :
05.         Para  $k = i + 1$  até  $\mathbf{A.tamanho}$ :
06.             Se  $\mathbf{A}[k]$  é melhor que  $\mathbf{A}[i]$ :
07.                  $\mathbf{temp} = \mathbf{A}[i]$ ;
08.                  $\mathbf{A}[i] = \mathbf{A}[k]$ ;
09.                  $\mathbf{A}[k] = \mathbf{temp}$ ;
10.      $\mathbf{A.tamanho} = N_E$ ;
11.     Para cada elemento  $k$  de  $\mathbf{B}$ :
12.          $\mathbf{B}[k].\text{aptidão} = \text{FO}(\mathbf{B}[k])$ ;
13.      $\mathbf{A} = \mathbf{A} + \text{Seleção}(\mathbf{B})$ ;

```

Figura 3.15: Algoritmo Genético elitista básico

Existem algumas vantagens na utilização do elitismo. Em primeiro lugar, pode permitir um incremento na taxa de convergência do GA. Em segundo lugar, impede a degradação da população ou a perda do melhor indivíduo ao longo das gerações. Porém existem também desvantagens: o elitismo pode aumentar exageradamente a pressão seletiva, possibilitando a convergência prematura. Em sistemas dinâmicos, onde o ótimo global pode variar sua posição ao longo do tempo, a otimização precisa ser continuamente realizada com uma considerável variação da população e também com uma pressão seletiva apropriada. Neste caso, o uso do elitismo pode não ser recomendável.

É possível substituir a seleção exclusivamente pelo elitismo, como de fato é feito em algumas variantes do GA, como por exemplo o algoritmo evolucionário CHC (*Cross generational elitist selection, Heterogeneous Recombination and Cataclysmic Mutation*) desenvolvido por

Eshelman [30]. O CHC é um algoritmo capaz de detectar possíveis convergências prematuras e reiniciar a busca conservando o melhor indivíduo e substituindo os demais por cópias mutantes do melhor. Dentre os vários mecanismos utilizados no CHC, destaca-se a seleção elitista.

A Fig. 3.16 exibe um GA simples que emprega seleção elitista. A população da geração corrente é expandida pelos filhos, gerados pela aplicação dos operadores de cruzamento e mutação. A proporção entre filhos e pais normalmente é feita de 1 : 1. A população da geração seguinte é construída eliminando os piores indivíduos da população expandida, de modo que retorne para o tamanho original. A população auxiliar **B**, criada com os filhos da população **A**, é incorporada à população **A** na linha 08 (processo representado pela “adição”). Um procedimento simples de ordenação por inserção é esboçado da linha 09 até a linha 14. Os melhores N_E cromossomos passam a ocupar o “topo” da população **A** expandida. Finalmente, através da atribuição realizada na linha 11, a população **A** é redimensionada para apenas N_E indivíduos do “topo”, concluindo uma geração do GA empregando a seleção elitista.

```

01. Cria uma população A aleatória;
02. Para cada elemento  $k$  de A:
03.      $\mathbf{A}[k].\text{aptidão} = \text{FO}(\mathbf{A}[k]);$ 
04. Para cada geração:
05.      $\mathbf{B} = \text{Mutação}(\text{Cruzamento}(\mathbf{A}));$ 
06.     Para cada elemento  $k$  de B:
07.          $\mathbf{B}[k].\text{aptidão} = \text{FO}(\mathbf{B}[k]);$ 
08.      $\mathbf{A} = \mathbf{A} + \mathbf{B};$ 
09.     Para  $j = 1$  até  $N_E$ :
10.         Para  $k = j + 1$  até  $\mathbf{A}.\text{tamanho}$ :
11.             Se  $\mathbf{A}[k]$  é melhor que  $\mathbf{A}[j]$ :
12.                  $\mathbf{aux} = \mathbf{A}[j];$ 
13.                  $\mathbf{A}[j] = \mathbf{A}[k];$ 
14.                  $\mathbf{A}[k] = \mathbf{aux};$ 
15.      $\mathbf{A}.\text{tamanho} = N_E;$ 

```

Figura 3.16: GA com seleção elitista

3.1.4 Definições de parâmetros e auto-adaptação

O GA possui uma certa quantidade de parâmetros cujos valores são críticos para o sucesso da heurística. O tamanho da população N_P , a probabilidade de cruzamento P_C , a probabilidade de mutação P_M (quando se utiliza codificação binária), o desvio da mutação σ_{Mk} (quando se utiliza codificação real), o número de indivíduos elitistas N_E , a pressão seletiva (definida pelos parâmetros do *ranking* ou pela capacidade do torneio). Uma combinação ideal de todos estes parâmetros permitirá ao GA obter a maior taxa de sucesso possível enquanto minimiza o número total de chamadas à FO.

3.1.4.1 O tamanho da população

O parâmetro N_P é um dos mais importantes, pois define a quantidade de chamadas à FO por geração do GA. Portanto, está diretamente associado ao desempenho computacional da heurística. Uma população muito grande permitiria uma exploração mais intensa do espaço de soluções, porém com uma convergência certamente mais lenta. Uma população muito pequena possibilitaria uma exploração muito pobre do espaço, favorecendo convergências prematuras. Ironicamente, populações muito pequenas são exploradas pelo micro-GA [31][32], fazendo com que o GA localize um ponto ótimo local. Ao fim da convergência, possivelmente prematura, a população é reciclada tendo seus indivíduos menos aptos substituídos por cromossomos aleatórios. O GA é executado mais uma vez com a população reciclada e este processo se repetirá até que nenhum outro ótimo (melhor) seja encontrado. A Fig. 3.17 exhibe o algoritmo simplificado para o micro-GA. Este se inicia criando, assim como nos outros algoritmos, uma população inicial **A** aleatória. Para cada ciclo, um GA é executado. Neste exemplo foi utilizado um com $N_E = 1$ (na linha 07, “Seleção” extrai $N_P - 1$ cromossomos de **B**), porém outros tipos de GA poderiam ser empregados. O critério de parada para o GA de cada ciclo foi ignorado no algoritmo da Fig. 3.17, mas normalmente utiliza-se um critério baseado na medida da perda de diversidade da população estimada pelo cálculo da variância dos genes correspondente entre todos os cromossomos da população. Para o exemplo da Fig 3.17, o critério de parada para do micro-GA baseia-se na comparação entre ótimo temporal da população **A**, representado por **A.melhor**, e o ótimo histórico armazenado na variável **aux**.

-
01. Cria uma população \mathbf{A} aleatória;
 02. Para cada ciclo:
 03. Para cada geração do ciclo:
 04. $\mathbf{B} = \text{Muta\c{c}\~{a}o}(\text{Cruzamento}(\mathbf{A}))$;
 05. Para cada elemento k de \mathbf{B} :
 06. $\mathbf{B}[k].\text{aptid\~{a}o} = \text{FO}(\mathbf{B}[k])$;
 07. $\mathbf{A} = \text{Sele\c{c}\~{a}o}(\mathbf{B}) + \mathbf{A}.melhor$;
 08. Se esta for a primeira gera\c{c}\~{a}o do primeiro ciclo:
 09. $\mathbf{aux} = \mathbf{A}.melhor$;
 10. Se $\mathbf{A}.melhor$ n\~{a}o for melhor do que \mathbf{aux} :
 11. Finaliza o micro-GA;
 12. Sen\~{a}o:
 13. $\mathbf{aux} = \mathbf{A}.melhor$;
 14. Preenche \mathbf{A} com $N_P - 1$ cromossomos aleat\~{o}rios;
 15. $\mathbf{A} = \mathbf{A} + \mathbf{aux}$;
-

Figura 3.17: Algoritmo simplificado para o micro-GA

O micro-GA troca o uso de popula\c{c}\~{o}es grandes por um n\~{u}mero maior de gera\c{c}\~{o}es. Alguns autores observaram vantagens nessa abordagem, uma vez que a defini\c{c}\~{a}o de N_P para o micro-GA parece ser mais simples de ser estabelecido do que para o GA convencional. De fato, a literatura cl\~{a}ssica, a cerca da predefini\c{c}\~{a}o do tamanho da popula\c{c}\~{a}o para o GA, est\~{a} firmemente associada com a hip\~{o}tese dos blocos de constru\c{c}\~{a}o (*Building-Block* – BB) [5]. Invariavelmente, em algum ponto a defini\c{c}\~{a}o de N_P necessitar\~{a} de algum conhecimento do espa\c{c}o de solu\c{c}\~{o}es, algo que pode ser inacess\~{i}vel para muitos problemas de engenharia. A express\~{a}o simplificada para N_P (para o GA convencional bin\~{a}rio) foi extra\~{i}da da literatura [5][33]:

$$N_P \approx 2^{L_{BB}} (0.7L_{BB} + \log N_{BB}) \quad (3.11)$$

Onde S_{BB} representa o comprimento m\~{e}dio dos BB; N_{BB} representa o n\~{u}mero de BBs. Determinar corretamente estes dois par\~{a}metros (especialmente o \~{u}ltimo) parece ser algo bastante abstrato para a maior parte dos problemas de engenharia concretos. Uma outra reinterpreta\c{c}\~{a}o da formula\c{c}\~{a}o de [5] mostrada a seguir extra\~{i}da de [33] parece ser mais acess\~{i}vel:

$$N_P \approx 2^{L_{BB}} \frac{\ell}{L_{BB}} \quad (3.12)$$

Onde ℓ representa o comprimento do cromossomo em número de bits. Para ilustrar a aplicação deste fórmula, algumas considerações interessantes podem ser formuladas de modo similar como em [33] para o caso do problema de minimização da função Rastrigin bidimensional. Considerando o comprimento total do cromossomo $\ell = 16$ bits e S_{BB} de aproximadamente 4 bits, resulta $N_P = 64$. Essa quantidade de cromossomos parece ser exagerada. De fato, 3.12 superdimensiona N_P , mas oferece um bom indício do parâmetro. Na prática muitos autores utilizam resultados estimados dessa forma como valores iniciais. O valor mais apropriado geralmente é refinado com base em experimentos.

Existem versões do GA, modificações que permitem determinar dinamicamente o número ideal de indivíduos. O princípio de funcionamento é o mesmo associado às equações (3.11) e (3.12): baseia-se na estimação de N_P de modo que o número de BBs alcance um valor suficiente para que a diversidade seja sustentada ao longo das gerações. Entretanto essa estimação é realizada dinamicamente durante a execução do GA pois nestas circunstâncias o modelo probabilístico pode ser realimentado por parâmetros autênticos extraídos diretamente do diagnóstico da diversidade da população, e não por vagas conjecturas [34]. Embora este GA modificado seja mais complexo, pode ser a única opção para a definição correta do tamanho da população para problemas extremamente complexos e computacionalmente onerosos.

Uma outra alternativa para o correto projeto de N_P é através do *Multi-Population GA* (MPGA). Várias populações concorrentes são sustentadas (não necessariamente sincronizadas) com valores de N_P diferentes. O processo funciona como um algoritmo evolucionário no topo, que tem como indivíduos da sua população as várias populações que formam o *cluster*. As populações conservam parâmetros distintos e as mais bem sucedidas fornecem indícios para o correto valor de N_P . Uma outra abordagem simplista seria apenas ignorar o problema de dimensionamento da população e utilizar a força bruta dos computadores de arquitetura paralela, pois o MPGA permite populações superdimensionadas, mesmo assim obtendo um desempenho computacional superior.

A não ser pela aplicação dos MPGA, a teoria clássica sobre o dimensionamento N_P está fortemente associada ao uso de alfabetos limitados. Para o dimensionamento do GA com codificação real é preciso estimar N_P considerando, por exemplo, um GA binário equivalente. Porém essa abordagem não considera a severa redução da dimensionalidade dos problemas obtida graças aos genes pontos-flutuantes. A teoria adequada para o GA real pode ser emprestada de outra escola de algoritmos evolucionários: as estratégias evolutivas (*Evolution Strategies* – ES).

Nascida na Europa em meados da década de 1960, as ES foram fundadas por P. Bienert, Ingo Rechenberg e Hans-Paul Schwefel na Universidade Técnica de Berlim. Na sua mais simples forma, uma ES se inicia com apenas uma solução candidata. A cada iteração, um filho é criado a partir da solução preexistente através, exclusivamente, da mutação por desvio gaussiano. A melhor solução dentre o pai e o filho sobreviverá para a próxima iteração. As soluções candidatas eram mantidas na sua forma analógica, ou seja, a codificação utilizada era normalmente a real. A partir do início dos anos de 1970 os principais teóricos dessa classe de algoritmos evolucionários começaram a estudar e publicar técnicas para o dimensionamento dos parâmetros das ES. Naquela época o único parâmetro de interesse era o desvio padrão da mutação gaussiana [25][29].

Na década de 1970 também se iniciaram os estudos sobre ES empregando não apenas 1 foco de busca, mas uma população de soluções candidatas capazes de se recombinar entre si através de operadores de cruzamento (a “recombinação” é de fato o termo mais empregado pelos teóricos das ES no lugar do termo “cruzamento”). A partir de então, os teóricos das ES tinham mais parâmetros com que se preocupar, todos eles bem familiares aos estudiosos dos algoritmos genéticos. Rapidamente se reconheceu a profunda relação existente entre o tamanho da população e o operador de recombinação. Em [35] são apresentados alguns desenvolvimentos interessantes que levam às equações evolutivas, que procuram prever alterações de diversidade ao longo das gerações. Algumas considerações envolvendo estas equações e alguns tipos comuns de operadores de cruzamento revelam inequações úteis associando o tamanho das populações ao parâmetro α do BLX e ao parâmetro η do cruzamento SBX, respectivamente:

$$\mu > \frac{3}{2\alpha(1 + \alpha) - 1} \quad (3.13)$$

e

$$\mu > 0.5(\eta + 3)(\eta - 1). \quad (3.14)$$

O parâmetro μ está relacionado com S_P , porém não de uma maneira direta. As ES com múltiplos indivíduos podem funcionar de modo bastante parecido com o GA com seleção elitista (Fig. 3.16). A cada iteração, a população se inicia com μ cromossomos e em seguida é expandida de λ indivíduos criados a partir de operações de recombinação e mutação. A população da geração seguinte é obtida através da seleção elitista, conservando os melhores μ indivíduos. As inequações (3.13) e (3.14) estabelecem as condições mínimas para que o operador de cruzamento possa conservar a diversidade das populações ao longo das gerações; foram obtidas considerando um modelo simples do algoritmo ES para $\lambda = \mu$ e cromossomos de apenas 1 gene. Considera-se que, como a população se duplica antes da aplicação da seleção, pode-se estabelecer que $S_P = 2\mu$. Considere como exemplo novamente o problema de

minimização da função Rastrigin bidimensional (Fig. 3.2). Para operador de cruzamento BLX com $\alpha = 0.5$, μ deve ser maior do que 6, o mesmo valor obtido para o operador de cruzamento SBX com $\eta = 3$. Portanto, S_P deve ser maior do que 12 e menor que 64 (este último valor que foi obtido utilizando (3.12)). Trata-se de um intervalo bastante largo cujo lapso reflete a dificuldade na definição de N_P sem quaisquer considerações sobre a complexidade do problema para o qual a heurística deverá ser aplicada. A precariedade das observações baseadas na equações (3.13) e (3.14) é também evidenciada pela ausência do parâmetro N_D (o número de parâmetros ou genes pontos-flutuantes) nas fórmulas.

As expressões (3.13) e (3.14) estabelecem que cruzamentos muito disruptivos (valor de α elevado e de η baixo) aumentarão o grau de diversidade, mesmo que o tamanho da população seja reduzido. O fenômeno se assemelha ao espalhamento de grãos sobre uma mesa: quanto mais espalhados e espaçados, menor será a quantidade necessária destes para cobrir toda a área. Quanto menos espalhados, maior a quantidade dos grãos necessária. Existem dois extremos: (1) poucos grãos muito espalhados, comportando-se tal como um algoritmo de busca aleatória. Cada novo filho seria radicalmente diferente dos pais devido ao cruzamento disruptivo, de tal modo que poderia ser considerado como um indivíduo completamente aleatório (a recombinação não teria significado). (2) muitos grãos pouco separados entre si, ou o equivalente com $\alpha = 0$ ou $\eta \rightarrow \infty$. A população seria infinita e colapsaria sobre o seu centro geométrico. Ambos os casos garantem a determinação do ótimo global do espaço, porém em condições absurdas onde uma quantidade infinita de tempo ou de memória seriam empregados. As inequações (3.13) e (3.14) simplesmente não distinguem essas situações extremas das factíveis, ou ainda entre quaisquer outras situações intermediárias.

3.1.4.2 Parâmetros de cruzamento

A discussão sobre a definição do parâmetro N_P revela a interdependência entre o operador de cruzamento e o tamanho da população para a conservação da diversidade da população de forma a evitar convergências em ótimos locais (convergência prematura). O cruzamento é capaz de criar indivíduos diferentes dos progenitores, acrescentando uma quantidade importante de inovação ao acervo genético da população. Devido sua importância, o operador de cruzamento tem sido muito pesquisado.

Porém existe um grande obstáculo para o estudo da definição estática ou automática dos parâmetros de cruzamento: a enorme quantidade de tipos de operadores disponíveis. Para a codificação binária existem basicamente 3 tipos: cruzamento de um ponto, multiponto e uniforme. Estes dois últimos tipos de cruzamento possuem um parâmetro adicional cada (além da probabilidade de cruzamento P_C): respectivamente o número de pontos de ruptura

e a probabilidade de ruptura. Para o cruzamento em cromossomos constituídos de valores reais, a quantidade de tipos de operadores é bem maior e muitos deles, como o BLX e o SBX, também possuem parâmetros de ajuste adicionais.

Portanto, o projeto dos mecanismos de cruzamento e de seus parâmetros começa na definição do próprio tipo de operador mais adequado a um determinado problema. Normalmente o tipo de cruzamento é determinado empiricamente, com o auxílio de exaustivos experimentos impraticáveis para alguns problemas computacionalmente ostensivos. Porém existem muitos estudos sobre mecanismos adaptativos e auto-adaptativos. Para o GA binário, a definição do tipo de cruzamento se concentra na pesquisa do número mais adequado e as posições dos pontos de ruptura. A maioria das abordagens são baseadas na teoria dos BB e procuram definir o número de pontos de cruzamento bem como as suas posições de modo a conservar importantes esquemas (combinações específicas de *bits* associadas a cromossomos de elevada aptidão). Estes processos adaptativos, embora bastante efetivos, possuem a desvantagem de exigir a detecção de associações entre *bits* e/ou entre esquemas, uma tarefa que pode aumentar a complexidade computacional da heurística.

Uma abordagem inovadora foi proposta por William M. Spears em 1995, onde o tipo de cruzamento é definido através de um gene (1 bit) extra nos cromossomos [36]. Trata-se, portanto, de uma técnica completamente auto-adaptativa, em que o próprio GA determina o cruzamento à medida que realiza sua busca. No momento do cruzamento, o bit extra de cada indivíduo é verificado. Se ambos possuem o bit extra igual a 0, o cruzamento de dois pontos será realizado; se ambos possuem o bit extra igual a 1, o cruzamento uniforme (com probabilidade de ruptura para cada gene de 50%) será realizado. Porém, se os bits extra forem diferentes, um ou outro tipo de cruzamento será aleatoriamente escolhido com iguais chances. Spears observou vantagens nesta abordagem em relação ao comportamento médio do GA devotado a um ou a outro tipo de cruzamento. É importante observar que os dois tipos de cruzamento são os extremos: o de dois pontos é muito pouco disruptivo, enquanto o uniforme é maximamente disruptivo. Observou-se que o GA escolhe conservar parte da reprodução baseada em um e parte no outro tipo de cruzamento simultaneamente, alcançando um equilíbrio populacional similar ao conseguido pelo cruzamento multiponto adequadamente sintonizado para o problema em questão e também para o estágio da otimização.

Os esquemas baseados na teoria dos BB ou na abordagem de Spears são destinados ao GA binário. Para os cromossomos reais muito mais estratégias de cruzamento existem, dificultando muito a elaboração de qualquer esquema adaptativo ou auto-adaptativo. Além disso, por exemplo, operadores como o BLX e o SBX podem ser equivalentes, dependendo dos parâmetros α e η . Portanto, a estratégia mais adequada parece ser especificar um tipo de cruzamento estaticamente e adaptar algum parâmetro disruptivo em função da diversidade da

população.

Porém, Manuel Lozano *et al* (2004) [37] apresentaram um tipo de cruzamento chamado de XHC (*Crossover Hill-Climbing*). A busca local *Hill-Climbing* é associada ao operador de cruzamento (o BLX) para obter um comportamento alegadamente auto-adaptativo. Para cada operação do XHC e para uma certa quantidade de interações, uma quantidade considerável de filhos são gerados, a partir de dois pais, através de repetidas chamadas ao cruzamento BLX. Os filhos gerados a cada iteração são comparados entre si e melhor dentre eles será escolhido para substituir o pior dentre os pais. Dentro do cenário construído em [37], o XHC é utilizado no interior de um SSGA (*Stead State GA*) que basicamente é um GA com seleção elitista empregando $N_P = N_E + 1$. O GA proposto utiliza um processo adaptativo que procura aplicar o XHC em uma pequena porcentagem dos casos, sendo o cruzamento convencional utilizado preferencialmente. Os autores alegam obter resultados melhores, maior confiabilidade e convergências em menos iterações, embora não seja discutido o custo computacional agregado.

A definição do parâmetro P_C é um pouco diferente das técnicas que atuam na escolha do tipo de cruzamento, pois é independente do tipo de codificação empregado. Uma abordagem auto-adaptativa integraria P_C como um gene adicional em cada cromossomo. Porém, para cada evento de cruzamento, a probabilidade P_C governante seria indeterminada, pois tanto a probabilidade de cruzamento do primeiro progenitor quanto a do segundo progenitor poderiam ser consideradas, além de algum tipo de combinação de ambas. Qualquer opção de regra para a escolha de P_C resultaria provavelmente muito mais complexa e difícil de justificar estatisticamente do que, por exemplo, a elegante técnica do bit extra de Spears. Por outro lado, uma importante técnica adaptativa já foi reportada na literatura: o *Adaptive GA* (AGA), que emprega probabilidades de cruzamento e mutação proporcionais, ou parcialmente proporcionais, à aptidão dos indivíduos. Quanto maior a aptidão menor P_C , pois é vantajoso proteger as boas soluções, enquanto submete-se as soluções menos aptas a cruzamentos, em prol da exploração de novas áreas do espaço de buscas. O parâmetro P_C seria dado por:

$$P_C = \begin{cases} \rho_{C1} \times (A_{max} - A_{melhor}) / (A_{max} - A_{med}) & \text{se } A_{melhor} \geq A_{med} \\ \rho_{C2} & \text{se } A_{melhor} < A_{med} \end{cases} \quad (3.15)$$

Onde ρ_{C1} e ρ_{C2} (usualmente $\rho_{C1} = \rho_{C2} = 1$) são constantes de controle; A_{max} representa a aptidão máxima da população; A_{melhor} representa a melhor dentre as aptidões dos progenitores; A_{med} representa a aptidão média da população. A desvantagem do processo é a dependência direta do controle de P_C dos valores de aptidão das soluções candidatas. Alguns problemas multiobjetivo podem utilizar vetores ao invés de escalares para representar a aptidão. Além disso, a distribuição não uniforme dos valores de aptidão podem fazer com que praticamente

apenas os valores extremos de P_C sejam utilizados. Para eliminar estes problemas, pode ser desejável colocar a população em ordem crescente de aptidão e utilizar uma espécie de *ranking* linear [38]:

$$P_C = \begin{cases} \rho_{C1} \times k_{melhor}/k_{med} & \text{se } k_{melhor} \geq k_{med} \\ \rho_{C2} & \text{se } k_{melhor} < k_{med} \end{cases} \quad (3.16)$$

Onde k_{melhor} representa o índice do melhor dentre os dois cromossomos progenitores; k_{med} representa o índice ou número do cromossomo que cuja aptidão mais se aproxima da média. De acordo com [38], a estratégia adaptativa para P_C melhora a convergência do GA, sendo o esquema dado pela equação (3.16) ligeiramente superior do que (3.15).

3.1.4.3 Parâmetros de mutação

Para cromossomos binários, existe apenas uma forma de mutação e apenas um parâmetro de controle, P_M , o qual governa um operador que transforma apenas 1 indivíduo. Nestas circunstâncias, parece ser mais viável conceber estratégias de controle adaptativas ou auto-adaptativas e, de fato, existem muitos interessantes trabalhos correlatos.

Como regra de definição estática, utiliza-se $P_M = 1/\ell$ [5]. Porém, isto não se compara a uma estratégia adaptativa ou auto-adaptativa, que é capaz de sintonizar P_M em função da própria procura, reduzindo as modificações dos cromossomos e refinando a busca à medida que estes se aproximam de algum ótimo.

Uma das estratégias mais simples é o DPC (*Dynamic Parameter Control*), onde P_M é deterministicamente decrementado em função do número da geração g . O primeiro trabalho relacionado foi publicado em 1989 por Forgy [5] e definia a probabilidade de mutação exclusivamente em função de g . Porém, empregava parâmetros empíricos obtidos para problemas bastante específicos. Um modelo mais geral foi proposto por Bäck e Schütz em 1996 [5], que desta vez considerava o comprimento do cromossomo ℓ :

$$P_M = \left[2 + \frac{\ell - 2}{G - 1}(g - 1) \right]^{-1} \quad (3.17)$$

onde G representa o número máximo de gerações e g , o número da geração corrente, se inicia em 1. É importante notar que para $g = 1$, $P_M = 0.5$ e $g = G$, $P_M = 1/\ell$.

A DPC não é verdadeiramente uma técnica adaptativa, visto que apenas modifica P_M de forma predefinida. É verdadeiramente superior à definição estática do parâmetro, porém não é capaz de lidar com eventuais “acidentes” ao longo do processo de busca. O espaço de

soluções pode ser altamente não-linear e não há garantias de que um decaimento exponencial de P_M seja satisfatório para quaisquer problemas. Portanto, uma técnica verdadeiramente adapta

Uma das mais simples e antigas técnicas adaptativas é a APC (*Adaptive Parameter Control*), inspirada na conhecida regra “1/5” de Rechenberg (1973) [25]. A regra foi originalmente criada para governar o valor do desvio de mutação σ_{Mk} dos cromossomos reais utilizados nas ES, porém pode ser adaptada para governar a probabilidade de mutação dos cromossomos binários. Ela estabelece que a proporção de mutações bem sucedidas deve ser de 1/5 ou 20% de todas as mutações realizadas em uma geração. Se a proporção de mutações bem sucedidas for inferior a 20%, P_M deverá diminuir; se for superior a 20%, P_M deverá aumentar. Matematicamente a regra assume a seguinte forma:

$$P_M^{g+1} = \begin{cases} \varsigma_M P_M^g & \text{se } \vartheta_M < 1/5 \\ P_M^g / \varsigma_M & \text{se } \vartheta_M > 1/5 \\ P_M^g & \text{se } \vartheta_M = 1/5 \end{cases} \quad (3.18)$$

onde os sobrescritos P_M^{g+1} e P_M^g representam respectivamente a probabilidade de mutação da geração seguinte $g + 1$ e da geração corrente g ; ς_M controla a taxa de incremento ou decréscimo da probabilidade de mutação, assumindo valores típicos entre 0.8 e 0.95; ϑ_M representa a razão entre o número de mutações bem sucedidas e o número de mutações totais. Como mutação bem sucedida deve-se entender aquela em que o cromossomo resultante adquire uma aptidão superior a do cromossomo original. É exatamente neste ponto onde se encontra a fraqueza desta técnica adaptativa: como determinar a razão de mutações bem sucedidas sem aumentar o custo computacional da heurística. De acordo com o algoritmo da Fig. 3.1, o cruzamento e a mutação são realizados em seqüência. Portanto, todo indivíduo submetido ao cruzamento precisaria ser avaliado duas vezes: antes da mutação e depois. Somente comparando as aptidões calculadas antes e depois seria possível determinar se a mutação em questão foi bem sucedida ou não. Considerando que P_C normalmente alcança taxas acima de 50%, isso implicaria no aumento do custo computacional na mesma proporção. Por outro lado, a taxa de sucesso ϑ_M poderia ser estimada a partir da pequena percentagem de indivíduos que sofrem mutação mas não o cruzamento.

Como alternativa adaptativa à regra de Rechenberg, existe estratégia utilizada no AGA para a probabilidade de mutação. O parâmetro P_M passa a ser inversamente proporcional à aptidão de cada indivíduo. Assim, melhores cromossomos serão sujeitos a modificações menos perturbadoras, permitindo a exploração mais cautelosa de locais do espaço de soluções provavelmente vizinhos de algum ótimo. A expressão que traduz matematicamente esta técnica

é praticamente idêntica à equação (3.16):

$$P_M = \begin{cases} \varsigma_{M1} \times k_{ind}/k_{med} & \text{se } k_{ind} \geq k_{med} \\ \varsigma_{M2} & \text{se } k_{ind} < k_{med} \end{cases} \quad (3.19)$$

onde ς_{M1} e ς_{M2} (usualmente $\varsigma_{M1} = \varsigma_{M2} = 0.5$) representam constantes de controle; k_{ind} representa o índice do indivíduo sujeito à mutação; k_{med} representa o índice do indivíduo de aptidão mais próxima da média.

A probabilidade de mutação também pode ser modificada através de uma abordagem auto-adaptativa, a SAPC (*Self-Adaptive Parameter Control*), em que P_M ocuparia um lugar nos cromossomos como um gene adicional. Esta abordagem foi inspirada na mutação auto-adapta

$$P_M^{g+1} = \left\{ 1 + \frac{1 - P_M^g}{P_M^g} \exp[-\tau_M N(0, 1)] \right\}^{-1} \quad (3.20)$$

onde P_M^{g+1} representa a probabilidade de mutação atualizada que deve governar o evento de mutação corrente; τ_M representa a “taxa de aprendizagem” cujos valores típicos situam-se em torno de 0.2.

Para o GA com codificação real, o parâmetro de interesse é o desvio σ_{Mk} . Desde o início da década de 1970 os principais teóricos, pais das modernas ES, já pesquisavam a respeito da definição do parâmetro de mutação σ_{Mk} . Praticamente todas as técnicas relacionadas provêm desta escola de algoritmos evolucionários ou são inspiradas em muitas de suas idéias.

A princípio, as ES não empregavam recombinação e por isso os valores definidos estaticamente para σ_{Mk} precisavam ser bastante elevados. Muito rápido se percebeu que valores de desvio variáveis ao longo das iterações poderiam melhorar enormemente a convergência dos algoritmos. Porém, o controle dinâmico não se mostrava muito vantajoso devido à ausência da recombinação, que tornava a convergência levada apenas pelo operador de mutação mais complexa e mais propensa a convergências prematuras. Por esse motivo as técnicas adaptativas já no princípio ganharam a atenção dos estudiosos.

A regra de 1/5 de Rechenberg, uma das primeiras técnicas adaptativas, foi notável pela simplicidade e aplicabilidade. Para o controle do desvio de mutação σ_M , utiliza-se exatamente a mesma equação (3.18), apenas substituindo P_M^g por σ_M^g (desvio corrente) e P_M^{g+1} por σ_M^{g+1} (desvio da próxima geração). O parâmetro σ_M^{g+1} é empregado em (3.8) ou em (3.9) no lugar de σ_{Mk} . A constante de controle ς_M , para este caso, ainda deve assumir valores entre 0.8 e 0.95, tipicamente 0.817.

No final da década de 1980 foram reportadas as primeiras experiências com uma técnica auto-adaptativa na qual o desvio padrão para a mutação gaussiana era colocado nos cromossomos como mais um gene. Na técnica auto-adaptativa isotrópica, para cada evento de mutação, o desvio σ_M^g deve ser extraído do cromossomo e atualizado para σ_M^{g+1} :

$$\sigma_M^{g+1} = \sigma_M^g \exp[\tau_M N(0, 1)] \quad (3.21)$$

onde σ_M^{g+1} governará a mutação em (3.9) como σ_{Mk} ($\sigma_{Mk} = \sigma_M \forall k$); τ_M neste caso representa a taxa de aprendizagem; $n(0, 1)$ representa uma amostra de uma variável aleatória de distribuição gaussiana, média nula e desvio padrão unitário.

A taxa de aprendizagem é um parâmetro muito importante para o processo auto-adaptativo e deve ser inversamente proporcional à raiz quadrada do número de dimensões do problema [25]. Muitos trabalhos procuraram determinar a constante de proporcionalidade que relaciona o número de dimensões à taxa de aprendizagem, porém cada problema parece possuir o seu próprio valor preferencial. Entretanto, para problemas difíceis e altamente multimodais recomenda-se $\tau_M = \sqrt{0.5/N_D}$, onde N_D representa o número de dimensões do problema [25].

Além da mutação auto-adaptativa isotrópica, a teoria sobre as ES celebra ainda a mutação anisotrópica, onde cada gene pode ser sujeito a um desvio diferente. Portanto, o cromossomo composto por $2M$ parâmetros: cada gene x_k associado a seu respectivo desvio σ_{Mk} . O objetivo da técnica auto-adaptativa anisotrópica é melhorar ainda mais a convergência da heurística, permitindo que o desvio espacial imposto pela mutação ao cromossomo seja independente nas dimensões do espaço de buscas, dado que a função objetivo também não varia obrigatoriamente da mesma forma em todas as dimensões. Porém, o aumento do número de dimensões devido a inclusão de variáveis extras pode ocasionar outros problemas à heurística como, por exemplo, a necessidade de uma população maior. Um caso extremo ocorre com outra técnica auto-adaptativa ainda mais aprimorada do que a anisotrópica: a mutação auto-adaptativa correlacionada. Nela, a covariância entre as diferentes variáveis de decisão x_k são codificadas também no interior do cromossomo. Isso resulta em ${}_{N_D}C_2$ (combinação de N_D 2 a 2) mais N_D variáveis extras. A adoção de tais técnicas mais sofisticadas de auto-adaptação deve ser cuidadosamente estudada, pois a necessidade do aumento populacional e do custo computacional decorrente podem ser intoleráveis para certos problemas [39], [25].

3.1.4.4 A pressão seletiva

Muitas das principais técnicas de seleção permitem o controle da pressão seletiva. Este controle pode ser definido estaticamente, permanecendo invariável ao longo das gerações, ou

pode ser controlado adaptativamente.

O controle da pressão seletiva é importante: uma pressão muito pequena (subdimensionada) pode permitir a convergência correta, porém, depois de uma (desnecessariamente) grande quantidade de gerações; uma pressão muito forte tende a eliminar rapidamente a diversidade da população, provocando a convergência prematura. Normalmente, a pressão seletiva é fixada, empregando um parâmetro estático determinado empiricamente com o auxílio de vários experimentos. Não raro recorre-se ao subdimensionamento, principalmente quando se dispõe de uma infraestrutura computacional robusta como um *cluster* de computadores.

Quando não é possível realizar experimentos devido ao tipo de problema e/ou pela falta de recursos computacionais, as técnicas adaptativas precisam ser consideradas seriamente. Existem basicamente dois tipos de abordagens: (1) estabelecendo regras de gestão da pressão seletiva em função da diversidade populacional; (2) estabelecendo procedimentos elaborados que procuram evitar que cromossomos similares sejam sistematicamente selecionados (valorização do material genético raro).

Um exemplo do primeiro tipo de abordagem é mostrado em [40], onde a regra de Rechenberg de $1/5$ é adaptada para o controle da pressão seletiva. A medida da diversidade é realizada de um modo bastante peculiar: através da medida da taxa de sucesso das operações de cruzamento-mutação, isto é, a quantidade de operações de cruzamento seguidas por mutação onde os filhos resultaram superiores ou equivalentes aos pais, dividida pela quantidade total de operações de cruzamento seguidas por mutação. Essa mesma taxa é inclusive utilizada como critério de parada do algoritmo, pois quando se anula significa que provavelmente a diversidade da população encontra-se irreversivelmente comprometida.

Um exemplo do segundo tipo de abordagem é mostrado em [41], onde cada indivíduo possui uma probabilidade de seleção proporcional à distância em relação a outro indivíduo de aptidão mais próxima. Em outras palavras, quanto mais raro for a ocorrência de um determinado valor de aptidão de um indivíduo, mais este será valorizado para a composição da geração seguinte. De um modo geral, indivíduos com aptidões mais difíceis de se atingir seriam sempre favorecidos.

É interessante observar a dificuldade para conceber alguma estratégia auto-adaptativa para o controle da pressão seletiva. Sendo um operador coletivo, a pressão seletiva codificada em algum cromossomo não poderia afetar a probabilidade de seleção de outro indivíduo diferente na mesma população. Mesmo assim, Eiben em trabalho um pouco mais recente [42] apresentou uma estratégia simples e criativa. Utilizando a seleção por torneio determinístico, a capacidade do subconjunto de seleção é codificada nos cromossomos como um gene adicional. Antes de submeter a população à seleção, cada um dos seus indivíduos é questionado

a respeito da pressão seletiva mais adequada. O processo é bastante parecido com uma eleição para a escolha da melhor pressão seletiva corrente. Como os demais genes dos cromossomos, a pressão seletiva é submetida à mutação, porém utilizando um esquema similar ao empregado em (3.20) para P_M . Nenhuma ressalva especial é feita em relação à recombinação, que trabalha sobre o gene adicional do mesmo modo como o faz com todos os restantes. Os resultados apresentados em [42] são animadores.

3.1.5 GA flexível

O GA envolve a aplicação sistemática de basicamente duas ações: a reprodução e a seleção. Entretanto estas ações podem ser efetuadas de inúmeras formas possíveis. Aplicando um dentre os incontáveis tipos de cruzamento, de mutação, que dependem do tipo codificação empregada. Aplicando um dentre os inúmeros métodos de seleção. O próprio GA possui uma fisiologia bastante variável. Por exemplo, é possível aplicar o elitismo independentemente da seleção ou é possível aplicar apenas o elitismo no lugar da seleção (seleção elitista).

A resposta para a pergunta de “qual deve ser a melhor receita para um tipo específico de problema” passa invariavelmente pela experimentação de várias técnicas, componentes e parâmetros diferentes. É relativamente simples permitir que o GA possa experimentar vários tipos de operadores genéticos, dado que estes são representados por funcionais dentro do algoritmo da Fig. 3.1. De fato, um funcional pode ser escolhido dentre vários outros de acordo com um simples parâmetro como, por exemplo “o tipo de cruzamento”. O mesmo se pode afirmar a respeito da função “Seleção”. Porém, alternar entre o GA elitista empregando, por exemplo, a seleção RW com *ranking* linear e um outro GA que emprega apenas a seleção elitista, como SSGA, não é algo direto. Para esta tarefa pode-se utilizar um GA tal como o mostrado na Fig. 3.18:

```

01. Crie uma população A aleatória;
02. Para cada elemento  $k$  de A:
03.      $\mathbf{A}[k].\text{aptidão} = \text{FO}(\mathbf{A}[k]);$ 
04. Ordene decrescente A;
05. Crie uma população auxiliar E de tamanho  $N_E$ ;
06. Para cada elemento  $k$  de E:
07.      $\mathbf{E}[k] = \mathbf{A}[k];$ 
08. Para cada geração:
09.      $k = 1;$ 
10.     Enquanto  $k < \mathbf{B}.tamanho$ :
11.          $\mathbf{aux} = \mathbf{A}[\text{mod}(k - 1, \mathbf{A}.tamanho) + 1];$ 
12.          $\mathbf{B}[k] = \text{Muta\c{c}\~{a}o}(\text{Cruzamento}(\mathbf{aux}, \mathbf{A}));$ 
13.          $\mathbf{B}[k].\text{aptidão} = \text{FO}(\mathbf{B}[k]);$ 
14.         Se  $\mathbf{B}[k]$  for melhor do que  $\mathbf{aux}$  ou  $u(0, 1) < P_B$  :
15.              $k = k + 1;$ 
16.     Para cada elemento  $k$  de A:
17.         Se  $\mathbf{A}[k].\text{idade} < \text{idade}_{max}$  ou  $\text{idade}_{max} < 0$ :
18.             Se  $\text{idade}_{max} > 0$ :
19.                  $\mathbf{A}[k].\text{idade} = \mathbf{A}[k].\text{idade} + 1;$ 
20.                  $\mathbf{B} = \mathbf{B} + \mathbf{A}[k];$ 
21.      $\mathbf{A} = \text{Sele\c{c}\~{a}o}(\mathbf{B});$ 
22.     Ordene decrescente A;
23.      $k = N_E;$ 
24.     Para  $i$  iniciando em  $N_E$  e enquanto  $i \geq 1$  e  $k \geq 1$ :
25.         Enquanto  $\mathbf{A}[i]$  for pior do que  $\mathbf{E}[k]$  e  $k \geq 1$ :
26.              $k = k - 1;$ 
27.             Se  $k \geq 1$ :
28.                  $\mathbf{E}[k] = \mathbf{A}[i];$ 
29.     Se elitista:
30.          $\mathbf{A} = \mathbf{A} + \mathbf{E};$ 

```

Figura 3.18: Algoritmo para o GA flexível

O algoritmo da Fig. 3.18 é diferente do anterior por fazer uso de uma população auxiliar

E (para os propósitos do elitismo) e um campo adicional nos cromossomos para o registro da idade dos mesmos. O GA se inicia de forma familiar, criando uma população principal aleatória **A**. Porém, a população **E** é criada na linha 05, constituída por N_E indivíduos, os quais serão, a princípio, cópias dos melhores indivíduos da população principal. Para isso, **A** não precisa ser completamente ordenada como sugere a linha 04: basta que os primeiros N_E indivíduos o sejam. Esta mesma observação se aplica à linha 18.

Dentro do *loop* principal do GA flexível, os operadores genéticos são aplicados sobre a população **A** gerando uma outra população auxiliar **B** totalmente constituída por indivíduos filhos. Porém, diferente do algoritmo anterior, onde as funções “Cruzamento” e “Mutação” manipulavam e retornavam populações inteiras, desta vez o procedimento é detalhado exibindo rotinas que processam e retornam um indivíduo por vez. Para tanto, o *loop*, que se inicia na linha 10 cria, na variável auxiliar **aux**, uma cópia de $\mathbf{A}[\text{mod}(k - 1, \mathbf{A.tamanho}) + 1]$ (linha 11), onde $\text{mod}()$ representa um operador que retorna o resto da divisão inteira do primeiro parâmetro pelo segundo e a propriedade *tamanho* retorna a capacidade do objeto população. Na linha 12 a função “Cruzamento” realiza a operação subentendida, considerando uma probabilidade P_C e tendo como indivíduos progenitores **aux** e pelo menos um indivíduo aleatoriamente escolhido da população **A**. Do modo como utilizada, a função “Cruzamento” retorna apenas um descendente por operação (um segundo descendente seria descartado) além de ser suficientemente geral a ponto de poder representar o cruzamento com mais de dois progenitores. O cruzamento com mais de um progenitor é uma interessante idéia proveniente da escola ES. De fato, em [42] o cruzamento real é apresentado como um procedimento que obtém cada gene do cromossomo filho através de uma combinação linear dos respectivos genes de cada indivíduo da população progenitora, uma generalização do cruzamento intermediário.

Caso o cruzamento não ocorra, a função “Cruzamento” retornará o próprio parâmetro **aux**. O indivíduo retornado é então submetido à mutação ainda na linha 12, a qual é realizada de acordo com uma probabilidade P_M . Similarmente ao cruzamento, caso a mutação não ocorra, o indivíduo retornado será uma cópia do próprio argumento. O valor objetivo do indivíduo descendente é calculado através da função “FO” na linha 13 e armazenado na propriedade *aptidão*. Em seguida, o indexador k é incrementado, mas condicionalmente apenas se o novo indivíduo for superior a **aux** ou se o operador $u(0, 1)$ retornar inferior a P_B , onde $u(0, 1)$ retorna um número aleatório de acordo com uma distribuição uniforme dentro de $[0, 1]$ e P_B representa a probabilidade de inserção do novo indivíduo na população **B**. Se $P_B = 1$, todos os indivíduos filhos serão automaticamente considerados. Para $P_B = 0$, apenas os novos indivíduos que forem superiores ao progenitor **aux** poderão constituir **B**. Para valores intermediários haverá uma preferência variável pelos indivíduos superiores ao ancestral. Este comportamento é inspirado no cruzamento XHC, embora realize apenas um passo de *hill*

climbing por indivíduo.

As linhas de 16 a 20 adicionam condicionalmente os indivíduos diretamente de **A** em **B**, desde que possuam idades inferiores a $idade_{max}$. Porém, caso $idade_{max} < 0$, toda a população **A** será incorporada a **B** incondicionalmente. Por outro lado, como todos os campos *idade* dos cromossomos são iniciados com zero, para $idade_{max} = 0$ nenhum cromossomo de **A** será incorporado a **B**. A população **A** é sobreposta pelos indivíduos selecionados de **B** na linha 21. As linhas de 20 a 24 substituem eventualmente os elementos da população auxiliar **E** por outros melhores originados da população principal **A**, conservando **E** ordenada. Caso o GA seja elitista, o código da linha 30 incorporará a população auxiliar **E** em **A**.

O elitismo pode se manifestar de duas formas possíveis no GA flexível. Na primeira, através da sobrevivência dos indivíduos da geração anterior graças à gestão do campo *idade* dos cromossomos. Na segunda forma, pela explícita captura dos melhores indivíduos da geração atual e a incorporação destes na geração subsequente através da população auxiliar **E**. Certamente a primeira forma é mais flexível, pois permite a limitação arbitrária do número de gerações que os elitistas possam sobreviver e gerar descendentes. Esta abordagem tem sido aplicada pela escola ES e tem permitido desempenhos atraentes em muitos casos. Entretanto, $idade_{max}$ passa ser mais um incômodo parâmetro que, como tantos outros, também precisa ser calibrado em função do problema. A segunda abordagem, realizada com a incorporação condicional de **E** à **A**, é mais familiar aos teóricos dos algoritmos genéticos. A necessidade de uma população auxiliar **E** pode ser adicionalmente vantajosa desde que contribua para uma avaliação estatística da convergência do GA, ou possa ser empregada para a reconstrução de novas populações entre os ciclos de um micro-GA baseado no GA flexível.

3.2 *Particle Swarm Optimization (PSO)*

O PSO foi introduzido em 1995 por J. Kennedy e R. C. Eberhart [6]. Foi descoberto através da simulação da dinâmica de sistemas como um bando de pássaros durante sua busca por alimentos, enxames de abelhas explorando ambientes, etc. O PSO é um algoritmo que simula uma comunidade (enxame) de agentes, extraíndo de seu coletivo toda a inteligência necessária para realizar otimizações ou sínteses. Dentro do espaço de buscas, os agentes são representados ou entendidos por partículas, que se movimentam de acordo com um conjunto restrito de regras mecânicas simples. As regras são parametrizadas por dados armazenados nas partículas, normalmente suas posições e velocidades.

O PSO tem se mostrado muito eficaz frente a problemas multimodais e/ou de alta dimensionalidade, a ponto de ser considerado superior aos GA/ES em diversas ocasiões. Muitos

autores classificam o PSO como mais um EA, dado que o enxame de partículas de fato evolui ao longo de iterações, buscando melhores posições no espaço. Outros negam o PSO como um EA por não promover a explícita sobrevivência dos mais aptos através do mecanismo da seleção. Neste trabalho o PSO será sempre considerado como um EA.

-
01. Cria um enxame de partículas **A** aleatórias;
 02. Para cada elemento k de **A**:
 03. $\mathbf{A}[k].\text{aptidão} = \text{FO}(\mathbf{A}[k]);$
 04. **global** = **A.melhor**;
 05. **B** = **A**;
 06. Para cada iteração:
 07. Para cada partícula k de **A**:
 08. $\mathbf{A}[k].\text{velocidade} = w \times \mathbf{A}[k].\text{velocidade} +$
 $u(0, 1) \times c_1 \times (\mathbf{B}[k].\text{posição} - \mathbf{A}[k].\text{posição}) +$
 $u(0, 1) \times c_2 \times (\text{global}.\text{posição} - \mathbf{A}[k].\text{posição});$
 09. $\mathbf{A}[k].\text{posição} = \mathbf{A}[k].\text{posição} + \mathbf{A}[k].\text{velocidade};$
 10. $\mathbf{A}[k].\text{aptidão} = \text{FO}(\mathbf{A}[k]);$
 11. Se **A**[k] for melhor do que **B**[k]:
 12. $\mathbf{B}[k] = \mathbf{A}[k];$
 13. Se **A.melhor** for melhor do que **global**:
 14. **global** = **A.melhor**;
-

Figura 3.19: Algoritmo para o PSO simples

A Fig. 3.19 mostra o algoritmo para o PSO simples. Primeiramente, uma população de partículas aleatórias **A** é criada e, em seguida, analisada através de recorrentes chamadas à FO no primeiro *loop* da linha 02 e 03. A partícula auxiliar **global** é criada na linha 04 com a função armazenar a melhor posição uma vez encontrada pela heurística. A população auxiliar **B**, a princípio, cópia de **A**, é criada na linha 05 com a função de armazenar a melhor posição já encontrada separadamente por cada partícula k de **A**. Para cada iteração, cada uma das partículas da população principal tem seus campos **velocidade** e **posição** atualizados pelas equações vetoriais que regem o movimento das partículas nas linhas 08 e 09. Após atualizada, cada partícula **A**[k] é reavaliada e, caso seja melhor do que a sua respectiva partícula memória **B**[k], será atribuída a esta. Ao fim de cada iteração, o melhor indivíduo de **A** pode ser atribuído a **global** caso o supere.

O campo *posição* das partículas é um vetor constituído pelas variáveis do problema,

portanto, indistinguível de um cromossomo empregado pelo GA com codificação real. Na linha 08, os parâmetros escalares w , c_1 e c_2 controlam respectivamente a influência da inércia, da melhor posição histórica encontrada pela partícula (memória da partícula ou memória local) e a influência da melhor posição histórica encontra por todo enxame (memória do enxame ou memória global) no movimento dos agentes. As parcelas multiplicadas por c_1 e c_2 são moderadas por valores aleatórios obtidos através da função $u(0, 1)$, que retorna uma amostra de uma variável aleatória uniforme entre 0 e 1. A Fig. 3.20 ilustra de forma bastante simplificada o processo vetorial de deslocamento das partículas dentro de um espaço bidimensional. A inércia aparece como sendo uma força de conservação da velocidade do agente, enquanto as partículas de memória local e global tentam alterar continuamente o curso do agente, funcionando como pontos de atração de intensidade variável.

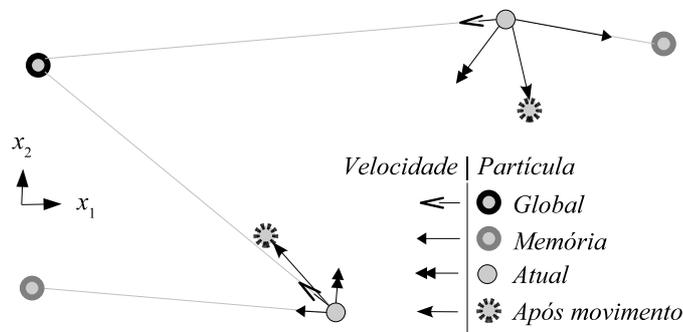


Figura 3.20: Articulação vetorial para o deslocamento das partículas no PSO.

A despeito do algoritmo do PSO parecer mais complexo do que o algoritmo genético mostrado na Fig. 3.1, todos os procedimentos e mecanismos estão explícitos na Fig. 3.19. Por outro lado, o GA empregava as funções “Cruzamento”, “Mutação” e “Seleção” na sua descrição, funções cujos algoritmos podem ser considerados tão sofisticados quanto todo o PSO. De fato, sua simplicidade é uma das principais vantagens em relação ao GA.

3.2.1 Definição dos parâmetros

Uma partícula dentro do PSO pode ser definida como $\mathbf{P} = \{\mathbf{V}, \mathbf{X}\}$, onde $\mathbf{V} = \{v_1, v_2, \dots, v_{k-1}, v_k, v_{k+1}, \dots, v_{N_D-1}, v_{N_D}\}$ representa o vetor velocidade, $\mathbf{X} = \{x_1, x_2, \dots, x_{k-1}, x_k, x_{k+1}, \dots, x_{N_D-1}, x_{N_D}\}$ representa o vetor posição e N_D representa o número de dimensões do problema. As equações de movimento usadas para atualizar uma partícula \mathbf{P} podem ser reescritas de forma escalar para cada componente como:

$$v_k^{t+1} = w v_k^t + u(0, 1)c_1(y_k - x_k^t) + u(0, 1)c_2(z_k - x_k^t) \quad (3.22)$$

e

$$x_k^{t+1} = x_k^t + v_k^{t+1} \Delta t \quad (3.23)$$

onde os subscritos indicam a dimensão e os sobrescritos indicam a iteração. Assim, v_k^{t+1} e x_k^{t+1} representam respectivamente as componentes da velocidade e da posição da partícula para a dimensão k e iteração $t+1$; v_k e x_k representam respectivamente as componentes da velocidade e da posição para a dimensão k e iteração corrente t . y_k e z_k representam respectivamente as componentes relacionadas à memória da partícula e à memória do enxame. Em (3.23) Δt representa o passo discreto de tempo, cujo valor é normalmente unitário, tal como empregado no algoritmo da Fig. 3.19.

Os pesos w , c_1 e c_2 controlam o processo de exploração do espaço de soluções. w define a massa da partícula: quanto maior, menor será a influência dos pontos de atração local e global, permitindo à partícula explorar amplamente o espaço de soluções. Os outros parâmetros, c_1 e c_2 , influenciam a partícula para que esta realize operações de busca local em torno dos possíveis ótimos atratores. O equilíbrio entre explorações locais e globais por parte dos agentes é fundamental para o sucesso desta heurística. Parta tanto, os valores normalmente empregados são $w = 1$, $c_1 = 2$ e $c_2 = 2$, determinados empiricamente [43]. Porém, a definição de limites para a velocidade também é fundamental para o processo. Normalmente, recomenda-se como máximo módulo da velocidade a diferença $x_k^{max} - x_k^{min}$, sendo x_k^{min} e x_k^{max} são respectivamente o mínimo e o máximo valor permitido para variável k do problema.

No início do processo de busca, valores de inércia maiores permitem explorar melhor o espaço. Porém, no final os pesos c_1 e c_2 tornam-se mais importantes, pois permitem o refinamento da busca em torno do suposto ótimo global. Portanto, um processo de adaptação dinâmico parece mais indicado para a definição dos pesos em função do número de iterações. De fato, em [43] é considerada a variação linear de w ao longo das iterações do PSO, por exemplo, de 0.9 a 0.2.

Uma outra abordagem seria a aplicação de alguma técnica adaptativa, criando algo como um APSO (*Adaptive* PSO). Em [44], cada uma das três parcelas referentes a cada um dos três pesos w , c_1 e c_2 , são somadas separadamente em etapas para todas as partículas na mesma iteração. Após a soma da parcela referente a inércia (w), todas as partículas são reavaliadas e, caso uma nova e melhor partícula *global* seja encontrada, w será incrementado de uma constante; caso contrário, w será decrementado de uma constante. O mesmo processo é então repetido para o peso c_1 e, em seguida, para c_2 . A constante utilizada nos incrementos e decrementos é normalmente 0.3, porém outros valores podem ser tolerados. O autor recomenda valores entre 0.2 e 5. É importante destacar que o processo executa sempre três vezes mais

chamadas da FO a cada iteração. Entretanto, neste artigo é mostrado, com auxílio de várias FO de teste, que a rápida convergência do APSO compensa o considerável aumento do custo computacional.

Seria possível conceber alguma espécie de estratégia auto-adaptativa para o PSO. Provavelmente seria como um PSO no topo de outro: o primeiro otimizaria os pesos das partículas enquanto o outro as variáveis do problema. Esta idéia não é absurda, entretanto o problema de definição de pesos é dinâmico, isto é, os valores ideais para os pesos variam em função do estágio da otimização e até mesmo da região do espaço onde a partícula se encontra. De acordo com alguns autores, o PSO pode mostrar alguma fraqueza para lidar com problemas dinâmicos, devido a velocidade de sua convergência e pela eventual irreversibilidade da contração do enxame de partículas. Em geral este aspecto é combatido através do re-semeamento de partículas aleatoriamente posicionadas próximas à periferia ou na vizinhança do enxame, como feito em [45] e [46]. Portanto, o PSO auto-adaptativo seria possível, porém talvez sofisticado a ponto de perder a sua vantajosa simplicidade em comparação ao GA.

Em 2002 Miranda e Fonseca [47] apresentaram uma abordagem auto-adaptativa, combinando o PSO com um ES. A heurística híbrida, chamada de *Evolutionary PSO* (EPSO), emprega uma ES para a definição dos pesos ao longo do processo de busca realizado pelas partículas. Não era a primeira vez que um híbrido era proposto, porém nenhuma outra abordagem anterior dedicava-se especificamente em promover a auto-adaptação resultando, ao mesmo tempo, em uma heurística consideravelmente robusta e simples.

Quanto ao número de partículas do enxame, na literatura não mais do que 40 partículas são empregadas mesmo para os problemas intensamente multidimensionais e multimodais. Os valores típicos oscilam entre 10 e 40 partículas e em média 20 são aplicadas para a maioria dos problemas reais. A definição deste parâmetro é sempre realizada empiricamente, não existindo quase nenhuma abordagem automática na literatura. Salva-se, entretanto, uma notável estratégia adaptativa batizada de TRIBES [48], que de fato é realmente capaz de determinar o tamanho do enxame adaptativamente entre outros parâmetros. Seu princípio de funcionamento baseia-se na criação de subconjuntos de partículas chamados de “tribos”. Cada tribo é sujeita a regras que determinam o seu crescimento ou decréscimo dependendo do seu desempenho em relação às outras tribos existentes.

3.2.2 Evolutionary PSO

O EPSO utiliza várias ES paralelas, uma para cada partícula do enxame. Através da mutação gaussiana, cada ES cria N_λ versões da sua respectiva partícula. Porém, a mutação não é aplicada às variáveis do problema, mas apenas aos pesos w , c_1 e c_2 , que passam a ser associados às partículas. Cada partícula com pesos mutantes e a própria partícula progenitora realizam seus movimentos, sendo em seguida avaliadas pela FO. A melhor partícula dentre as mutantes e a progenitora será escolhida para tomar lugar como a próxima progenitora na iteração seguinte. A Fig. 3.21 exhibe o algoritmo básico para o EPSO:

```

01. Cria um enxame de partículas A aleatórias;
02. Para cada elemento  $k$  de A:
03.      $\mathbf{A}[k].\text{aptidão} = \text{FO}(\mathbf{A}[k]);$ 
04. global = A.melhor;
05. B = A;
06. Para cada iteração:
07.     Para cada partícula  $k$  de A:
08.         aux1 =  $\mathbf{A}[k];$ 
09.          $\mathbf{A}[k] = \text{Movimente}(\mathbf{A}[k]);$ 
10.          $\mathbf{A}[k].\text{aptidão} = \text{FO}(\mathbf{A}[k]);$ 
11.         Repita  $N_\lambda$  vezes:
12.             aux2 =  $\text{Movimente}(\text{Muta\c{c}\~{a}\~{o}}(\mathbf{aux1}));$ 
13.             aux2.aptidão =  $\text{FO}(\mathbf{aux2});$ 
14.             Se aux2 for melhor do que  $\mathbf{A}[k]$ :
15.                  $\mathbf{A}[k] = \mathbf{aux2};$ 
16.             Se  $\mathbf{A}[k]$  for melhor do que B:
17.                  $\mathbf{B}[k] = \mathbf{A}[k];$ 
18.         Se A.melhor for melhor do que global:
19.             global = A.melhor;

```

Figura 3.21: Algoritmo para o EPSO simples

Dentro do EPSO, uma partícula pode ser definida como $\mathbf{P} = \{\mathbf{V}, \mathbf{X}, w, c_1, c_2, c_3\}$, ou seja, os pesos são incorporados à partícula. O procedimento “Movimente” do algoritmo da Fig. 3.21 atualiza apenas os elementos \mathbf{V} e \mathbf{X} da partícula, porém utilizando uma outra versão para (3.22):

$$v_k^{t+1} = w^{t+1} v_k^t + c_1^{t+1} (y_k - x_k^t) + c_2^{t+1} [z_k + c_3 n(0, 1) - x_k^t]. \quad (3.24)$$

Onde os sobrescritos indicam a iteração a que pertencem os respectivos parâmetros. O peso adicional c_3 representa o desvio padrão do ruído gaussiano aditivo ao qual a posição da partícula global é submetida. É importante observar a ausência das variáveis aleatórias uniformes representadas por $u(0, 1)$ como em (3.22).

O procedimento “Mutação”, por outro lado, lida apenas com os pesos, atualizando-os de w_t , c_1^t e c_2^t para w^{t+1} , c_1^{t+1} e c_2^{t+1} respectivamente, através de:

$$\{w^{t+1}, c_1^{t+1}, c_2^{t+1}, c_3^{t+1}\} = \{w^t, c_1^t, c_2^t, c_3^t\} + \tau_P n(0, 1) \quad (3.25)$$

onde τ_P representa o desvio padrão para o ruído gaussiano aditivo utilizado na atualização dos pesos, tipicamente assumindo valor da ordem de 0.1. Além do ruído gaussiano aditivo, em [49] é mostrado que a atualização dos pesos através de um ruído multiplicativo log-normal pode proporcionar um desempenho superior ao EPSO:

$$\{w^{t+1}, c_1^{t+1}, c_2^{t+1}, c_3^{t+1}\} = \{w^t, c_1^t, c_2^t, c_3^t\} \times \exp[\tau_P n(0, 1)]. \quad (3.26)$$

O valor típico para τ_P para a atualização multiplicativa log-normal também é de aproximadamente 0.1.

No algoritmo da Fig. 3.21, cada partícula k do enxame \mathbf{A} ($\mathbf{A}[k]$) é copiada para uma partícula chamada de “**original**” antes de ser submetida ao movimento e à avaliação. Todas as partículas mutantes são criadas como cópias de **original**, porém são submetidas à mutação dos pesos, ao movimento, avaliação e comparação com a partícula $\mathbf{A}[k]$. Caso a partícula **mutante** seja melhor do que $\mathbf{A}[k]$, a **mutante** tomará o lugar de $\mathbf{A}[k]$. Este processo é claramente a seleção do melhor movimento, que provavelmente estará associado ao melhor conjunto de pesos.

De acordo com [47] e [49], o EPSO é bem sucedido como heurística auto-adaptativa, demonstrando melhores resultados e maior confiabilidade do que o PSO para muitos problemas baseados em várias funções de teste diferentes. Uma potencial desvantagem seria o aumento da quantidade de chamadas da FO por iteração. Entretanto, a multiplicação de partículas dentro do processo auto-adaptativo aumenta também a quantidade de pontos de exploração do espaço. Também o processo auto-adaptativo em si garante uma convergência mais rápida em menos iterações, possivelmente compensando o aumento do custo computacional.

3.2.3 Topologias de cooperação entre partículas

O PSO tal como descrito no algoritmo da Fig. 3.19 e Fig. 3.21, emprega a posição da partícula global nas expressões do movimento para representar a influência do enxame sobre os agentes. Este é o tipo mais simples de topologia, conhecido como topologia em estrela. De acordo com [50], esta abordagem não é recomendável para problemas altamente multimodais. Para estes problemas recomenda-se topologias em que a melhor partícula local ocupa o lugar da melhor global como referência para o enxame. Em [50] um PSO “completamente informado” é proposto, em que cada partícula utiliza uma combinação linear de todas as partículas memória individuais do enxame no lugar da partícula global. Em [51] um sistema hierárquico coloca as partículas gravitando em torno de partículas locais de aptidão mais elevada, que por sua vez gravitam em torno de outras partículas de aptidão ainda maior e assim por diante. Estratégias como estas permitem conservar a diversidade do enxame por mais tempo, evitando convergências prematuras motivadas pela rápida contração do enxame (fenômeno conhecido como *crowding*).

Miranda e Naing [49] apresentaram em 2005 uma topologia alternativa adaptada sobre o EPSO, chamada de estrela estocástica. Esta abordagem simplesmente omite intermitentemente a parcela associada com a contribuição da partícula global da equação (3.24) em função de uma probabilidade P_Z :

$$v_k^{t+1} = \begin{cases} w^{t+1} v_k^t + c_1^{t+1}(y_k - x_k^t) + c_2^{t+1}[z_k + c_3 N(0, 1) - x_k^t] & \text{se } u(0, 1) \leq P_Z \\ w^{t+1} v_k^t + c_1^{t+1}(y_k - x_k^t) & \text{se } u(0, 1) > P_Z \end{cases} \quad (3.27)$$

De acordo com [49], $P_Z = 0.2$ garantiu resultados consideravelmente superiores ao obtido pelo EPSO original utilizando como FO uma função de Rosenbrock em 30 dimensões. O mesmo trabalho também aponta vantagens da aplicação da atualização multiplicativa log-normal dos pesos ao invés da atualização aditiva gaussiana.

3.2.4 Observação das restrições

Assim como o GA real, o PSO também pode necessitar de alguma estratégia de truncagem do espaço de soluções. Seguindo a metáfora dinâmica para o movimento das partículas, sem quaisquer regras que as impeçam, as partículas simplesmente atravessarão os limites de cada dimensão livremente. Existem basicamente três abordagens para prevenir que esse fenômeno ocorra ou que possa prejudicar o funcionamento da heurística de alguma forma [43].

A primeira abordagem faz uso de fronteiras refletoras, em que o sentido do movimento da partícula numa dada dimensão se inverte no momento em que esta colide com uma das duas fronteiras desta mesma dimensão. A segunda abordagem implica no uso das fronteiras absorvedoras, em que o módulo da velocidade numa dada dimensão se anula no momento que a partícula colide com a fronteira. A última abordagem faz uso de fronteiras transparentes. Simplesmente as partículas podem atravessar os limites de uma dada dimensão livremente. Porém, uma vez fora destes limites, a FO não será mais aplicada sobre a partícula, ficando a sua aptidão desatualizada. Dessa forma, os atratores local e global eventualmente arrastarão a partícula de volta ao intervalo válido.

A Fig. 3.22 ilustra o processo de observação das restrições: fronteiras refletoras (a), absorvedoras (b) e transparentes (c). Embora para caso (c) a partícula possa violar as restrições do problema, os pontos de atração certamente a trarão de volta. Enquanto isso, a solução candidata por ela representada não poderá ser considerada válida.

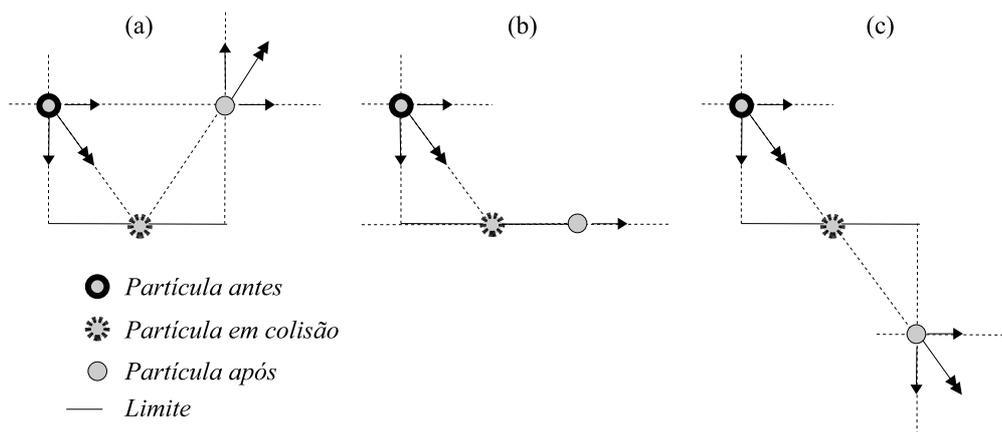


Figura 3.22: Três abordagens para a restrição dos parâmetros no PSO: fronteiras refletoras (a), absorvedoras (b) e transparentes (c).

De acordo com [43], fronteiras transparentes proporcionam resultados ligeiramente superiores do que os geralmente obtidos através das outras duas abordagens.

3.2.5 PSO flexível

Define-se como flexível um PSO cuja implementação permita a aplicação de técnicas diversificadas sem mudanças na sua estrutura básica, a exemplo do GA flexível apresentado na seção 3.1.5, capaz imitar tanto a fisiologia de uma ES quanto a de um GA canônico. Porém, para o PSO, a literatura não é tão polarizada quanto para os EA. Existem muitas

estratégias ou variantes exóticas, o que torna impossível projetar um algoritmo polivalente capaz de opcionalmente simular todas as principais abordagens disponíveis. Entretanto, um bom candidato para o PSO flexível seria uma adaptação do EPSO.

O EPSO é auto-adaptativo, porém, modificações modestas na sua fisiologia permitem que se comporte exatamente como um PSO canônico. Outro aspecto que pode ser explorado é a topologia de cooperação. O EPSO pode opcionalmente empregar a topologia em estrela estocástica, porém não é difícil adaptá-la para uma topologia de cooperação *local* estocástica, em que uma partícula localmente superior substitui a partícula global. Evidentemente fazendo $P_Z \rightarrow 1$ resultaria nas topologias estrela ou local convencionais. Adaptá-la para uma topologia de cooperação *local* estocástica, em que uma partícula localmente superior substitui a partícula global. Evidentemente fazendo $P_Z \rightarrow 1$ resultaria nas topologias estrela ou local convencionais.

Uma outra interessante modificação seria o re-semeamento de partículas aleatórias em determinadas circunstâncias, por exemplo, quando o enxame se concentra comprometendo sua diversidade. Tal procedimento é comparável ao mecanismo de ciclos empregado pelo micro-GA. Esta estratégia poderia ser efetivada, por exemplo, reiniciando aleatoriamente 50% das partículas menos aptas quando a diversidade regredisse até um determinado valor. A diversidade pode ser definida do mesmo modo que em [46], isto é, proporcional à distância euclidiana média das partículas em relação ao centro geométrico do enxame:

$$Diversidade = \frac{1}{S_P} \sum_{j=1}^{S_P} \sqrt{\sum_{k=1}^{N_D} \left(\frac{\bar{x}_k - x_{jk}}{x_k^{max} - x_k^{min}} \right)^2} \quad (3.28)$$

onde

$$\bar{x}_k = \frac{1}{S_P} \sum_{j=1}^{S_P} x_{jk} \quad (3.29)$$

Quando a diversidade torna-se inferior a um valor convencionalizado, o re-semeamento acontece. Evidentemente, formas alternativas para a estimação da diversidade poderiam ser menos dispendiosas computacionalmente. Por exemplo, substituindo a distância euclidiana da partícula pela máxima diferença entre as componentes da sua posição e a posição do centro geométrico do enxame:

$$Diversidade = \frac{1}{S_P} \sum_{j=1}^{S_P} \max \left\{ \frac{\bar{x}_1 - x_{j1}}{x_1^{max} - x_1^{min}}, \frac{\bar{x}_2 - x_{j2}}{x_2^{max} - x_2^{min}}, \dots, \frac{\bar{x}_{N_D} - x_{jN_D}}{x_{N_D}^{max} - x_{N_D}^{min}} \right\} \quad (3.30)$$

A Fig. 3.23 exibe um possível algoritmo para o PSO flexível que se inicia de modo bas-

tante parecido aos algoritmos PSO e EPSO anteriormente mostrados. As maiores diferenças entre este algoritmo e os anteriores começam no *loop* interior, onde a partícula global é definida (linha 07). Se a topologia for local, as linhas de 08 a 10 devem procurar por um indivíduo da população \mathbf{B} que seja imediatamente superior a $\mathbf{B}[k]$. Se a topologia for global, a respectiva partícula deverá ser definida como o melhor indivíduo da população \mathbf{B} . Este processo é dependente de uma ordenação do enxame decrescente em função da aptidão da população \mathbf{B} , o que é garantido no algoritmo do PSO flexível através das linhas 21 a 23. De fato, o processo de ordenação torna-se similar ao algoritmo *bubble sort* [52] e completa-se após várias iterações, porém a ordenação é sistematicamente conservada subsequente. O efeito obtido é similar à propagação dos melhores indivíduos em direção do topo da população, um processo inspirado no HPSO [51].

```

01. Cria um enxame de partículas A aleatórias;
02. Para cada elemento  $k$  de A:
03.      $\mathbf{A}[k].\text{aptidão} = \text{FO}(\mathbf{A}[k]);$ 
04. B = A;
05. Para cada iteração:
06.     Para cada partícula  $k$  de A:
07.         global = B.melhor;
08.         Se topologia for local e  $k > 1$ :
09.             Decremente  $j$  de  $k - 1$  até 1 enquanto B[ $j$ ] não for melhor que B[ $k$ ];
10.             Se B[ $j$ ] for melhor que B[ $k$ ]: global = B[ $j$ ];
11.         aux2 = A[ $k$ ];
12.         A[ $k$ ] = Movimente(A[ $k$ ]);
13.         A[ $k$ ].aptidão = FO(A[ $k$ ]);
14.         Repita  $N_\lambda$  vezes:
15.             aux3 = Movimente(Mutação(aux2));
16.             aux3.aptidão = FO(aux3);
17.             Se aux3 for melhor do que A[ $k$ ]:
18.                 A[ $k$ ] = aux3;
19.             Se A[ $k$ ] for melhor do que B[ $k$ ]:
20.                 B[ $k$ ] = A[ $k$ ];
21.             Se  $k > 1$ :
22.                 Se B[ $k$ ] for melhor do que B[ $k - 1$ ]:
23.                     Permute A[ $k$ ] e A[ $k - 1$ ], permute B[ $k$ ] e B[ $k - 1$ ];
24.             Se a Diversidade(A) < mínimo:
25.                 Para  $k$  iniciando em A.tamanho/2 até A.tamanho:
26.                     B[ $k$ ] = A[ $k$ ] = “Nova partícula aleatória”;
27.                     B[ $k$ ].aptidão = A[ $k$ ].aptidão = FO(A[ $k$ ]);

```

Figura 3.23: Algoritmo para o PSO flexível

A função “Movimente” (linhas 12 e 15 do PSO flexível) aplica opcionalmente a equação de atualização da velocidade (3.27) ou a seguinte equação:

$$v_k^{t+1} = \begin{cases} w^{t+1} v_k^t + c_1^{t+1} u(0, 1)(y_k - x_k^t) + c_2^{t+1} u(0, 1)[z_k + c_3 n(0, 1) - x_k^t] & \text{se } u(0, 1) \leq P_Z \\ w^{t+1} v_k^t + c_1^{t+1} u(0, 1)(y_k - x_k^t) & \text{se } u(0, 1) > P_Z \end{cases} \quad (3.31)$$

Que se diferencia de (3.27) apenas pela presença dos fatores $u(0, 1)$ nas parcelas referentes aos coeficientes c_1 e c_2 . Fazendo $c_3 = 0$ e $P_Z = 1$, (3.31) torna-se idêntica à equação de atualização da velocidade para o PSO canônico.

Também subentende-se que a função “Mutação” (linha 15) é capaz de empregar opcionalmente a mutação gaussiana, log-normal ou, caso deseje-se um comportamento idêntico ao do PSO canônico, nenhuma mutação.

A função “Diversidade” na linha 24 aplica opcionalmente o cálculo definido pelas equações (3.28) e (3.29) ou por (3.28) e (3.29).

Capítulo 4

GA e PSO aplicados à síntese de FBG

4.1 Introdução

Heurísticas como o GA e o PSO têm sido aplicadas com sucesso para a síntese ou otimização de filtros ópticos. De fato, o problema de síntese de FBGs possui características que praticamente impossibilitam a exclusiva aplicação de técnicas de otimização clássicas.

Entretanto, para conseguir aplicar o GA ou o PSO na síntese de FBGs, antes é necessário determinar quais parâmetros e de que modo devem ser mapeados no cromossomo ou nas partículas, para permitir a máxima eficácia da heurística¹. Em segundo lugar, é necessário definir a FO.

Este capítulo tem como objetivo apresentar um conjunto de métodos e práticas capazes de conectar a técnica de análise matricial mostrada no Capítulo 2 às heurísticas apresentadas no Capítulo 3. Será mostrado como o problema de síntese de FBGs pode ser mapeado nos cromossomos ou partículas, desde o mapeamento simples até técnicas mais elaboradas que permitem a redução do espaço de buscas e o aumento da eficácia da heurística. A definição da FO também será mostrada neste capítulo, com especial ênfase nas técnicas de amostragem econômicas capazes de reduzir o custo computacional da heurística. Uma destas técnicas de amostragem, quando combinada com algumas modificações simples da heurística, permite oferecer considerável ganho computacional à otimização.

¹Neste capítulo “heurística” refere-se exclusivamente ao GA e PSO.

4.2 Decodificação

Como mostrado no Capítulo 2, a otimização opera sobre os parâmetros das seções uniformes: $\delta n_{eff i}$, ν_i , Λ_i e Δz_i , suficientes para representar uma FBG. Adicionalmente, podem ser considerados os deslocamentos de fase discretos entre as seções ϕ , somando ao todo 5 parâmetros por seção. Assim, uma FBG de M seções pode ser representada pelo vetor

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} \quad (4.1)$$

onde os subvetores \mathbf{x}_1 , \mathbf{x}_2 , \mathbf{x}_3 , \mathbf{x}_4 e \mathbf{x}_5 são:

$$\begin{aligned} \mathbf{x}_1 &= \{ \delta n_{eff 1}, \delta n_{eff 2}, \dots, \delta n_{eff M} \} \\ \mathbf{x}_2 &= \{ \nu_1, \nu_2, \dots, \nu_M \} \\ \mathbf{x}_3 &= \{ \Lambda_1, \Lambda_2, \dots, \Lambda_M \} \\ \mathbf{x}_4 &= \{ \Delta z_1, \Delta z_2, \dots, \Delta z_M \} \\ \mathbf{x}_5 &= \{ \phi_1, \phi_2, \dots, \phi_M \} \end{aligned} \quad (4.2)$$

O vetor \mathbf{X} pode ser utilizado diretamente como cromossomo (codificação real) ou traduzido como uma seqüência binária (codificação binária). Independente do tipo de codificação ou do comprimento do alfabeto, a dimensionalidade do problema será função de M . Desde que a condição $\Delta z_i \gg \Lambda_i$ (usualmente faz-se $\Delta z_i \equiv 100\Lambda_i$) seja satisfeita [16], o cálculo do espectro de refletividade da FBG representada por \mathbf{X} será tanto mais preciso quanto maior o número de seções. Assim sendo, M forçosamente assume valores da ordem de centenas para grades com comprimentos típicos da ordem dos centímetros. Como discutido no Capítulo 3, de acordo com a literatura básica, esta alta dimensionalidade pode demandar do GA populações da ordem de milhares de indivíduos, valores impraticáveis para o problema em questão. O impacto sobre o PSO precisaria ser avaliado, mas provavelmente seria tão indesejável quanto para o GA. Dentre todas as meta-heurísticas o GA binário seria o menos indicado, por aumentar ainda mais a dimensionalidade do problema.

4.2.1 Decodificação normalizada

Teoricamente a codificação real não exige a definição de intervalos de validade para os parâmetros tal como ocorre para a codificação binária. Entretanto, na prática os parâmetros relacionados em (4.2) encontram-se inseridos entre rígidos intervalos relacionados tanto com a viabilidade dos processos de fabricação de FBGs como também com a técnica numérica

matricial de análise disponível. A máxima variação de índice de refração induzida observável obtida através da fotossensibilidade possui o limite máximo prático de 10^{-2} . Para FBGs gravadas em sílica dopada, não é possível encontrar o parâmetro δn_{effi} nem tão pouco o produto $\nu_i \times \delta n_{effi}$ superiores ao referido valor de máximo. Aceita-se, portanto, o intervalo $[0; 10^{-2}]$ como os limites absolutos para δn_{effi} e $[0; 10^{-2}/\delta n_{effi}]$ para ν_i . Entretanto, valores típicos para a variação de índice de refração induzida não superam 10^{-3} , dado que FBGs com $\nu_i \times \delta n_{effi} = 10^{-3}$ atingem valores de refletividade de 100% com comprimentos inferiores a 1cm. O parâmetro Λ , por outro lado, está associado com a banda de comprimento de onda desejada para o projeto, variando tipicamente dentro do intervalo $[(\lambda_0 - \Delta\lambda)/(2n_{eff}); (\lambda_0 + \Delta\lambda)/(2n_{eff})]$, onde λ_0 representa o comprimento de onda central para a banda de resposta do filtro e $\Delta\lambda$ define sua largura espectral (normalmente $\Delta\lambda$ é inferior a 0.5% de λ_0).

O parâmetro Δz_i é mais limitado pelo modelo de FBG do que pelas limitações tecnológicas de manufatura de grades. O uso do seccionamento uniforme discreto é apenas um artifício do modelo numérico, não está associado com a anatomia das FBGs. A técnica de gravação ponto-a-ponto, por exemplo, pode criar apodizações arbitrárias e suaves, que são desprovidas de seções discretas. Pode-se entender que as grades fabricadas por tal processo sejam constituídas por seções com Δz_i infinitesimais, ou melhor, do mesmo comprimento que Λ_i (por definição, o menor valor possível). Embora esta técnica ainda possa criar FBGs uniformes de comprimentos teoricamente ilimitados, existem dificuldades práticas para a criação de grades muito longas (da ordem de metros). Outra técnica de gravação, através de máscaras de fase, também permite apodizações suaves acompanhando, obviamente, o padrão gravado nas máscaras. É mais comum, entretanto, o emprego de padrões uniformes. Portanto, o comprimento mínimo da seção também é Δz_i (assim como para a técnica ponto-a-ponto); o comprimento máximo é algo como o comprimento da própria máscara. Uma variante da técnica por máscaras utiliza janelamento para conseguir o controle da apodização da grade mesmo empregando máscaras uniformes. Este processo (e possíveis variantes) é o único que materializa o modelo das seções uniformes. O processo expõe a fibra a trechos da máscara de fase isolados. O comprimento mínimo das seções, da ordem de 0.5mm, está associado com problemas relacionados com a difração da luz UV nas bordas de uma fenda aberta em um anteparo empregado para bloquear parcialmente a luz sobre a máscara. O valor de Δz_i máximo é da ordem do comprimento da própria máscara, isto é, da ordem de centímetros. Por outro lado, A técnica de análise matricial baseada da teoria dos modos acoplados (mostrada ligeiramente no Capítulo 2) exige que $\Delta z_i \gg \Lambda_i$. Em [16] é sugerido a relação $\Delta z_i \geq 100 \times \Lambda_i$, o que implica em um valor mínimo para Δz_i de $100 \times \Lambda_i$.

O parâmetro ϕ , por outro lado, raramente assume valores diferentes de zero, e apenas o faz para projetos específicos de FBGs onde poucas seções podem apresentar valores signi-

ficativos. A ocorrência de um valor de $\phi \neq 0$ em uma das seções de uma FBG, como mostrado no Capítulo 2, provoca uma fenda na banda de reflexão com algumas características bastante previsíveis. Como trata-se de ângulo, este parâmetro efetivamente pode possuir qualquer valor numérico e finito, não existindo valores limites, muito embora qualquer ângulo seja equivalente a algum pertencente ao intervalo $[0; 2\pi]$ ou $[-\pi; \pi]$.

Percebe-se que os diversos parâmetros associados ao projeto de uma FBG estão inseridos dentro de limites contrastantes. Os parâmetros associados aos procedimentos de mutação no GA e de movimentação de partículas no PSO precisam considerar estes limites no momento que os cromossomos ou as partículas são atualizados ou modificados. Por exemplo, é convencional normalizar o parâmetro σ_M de modo que o mesmo valor possa ser aplicado em diferentes parâmetros na mutação isotrópica. Uma alternativa interessante é fazer justamente o oposto: normalizar os elementos de \mathbf{x}_k ($k = \{1, 2, 3, 4, 5\}$), em função dos respectivos intervalos. Assim:

$$\mathbf{x}_k(i) = x_k^{min} + (x_k^{max} - x_k^{min}) \times \mathbf{y}_k(i), \quad (4.3)$$

onde o intervalo $[x_k^{min}; x_k^{max}]$ representa os limites para o subvetor \mathbf{x}_k da FBG \mathbf{X} . O vetor $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4, \mathbf{y}_5\}$ compõe diretamente o cromossomo ou o vetor posição de uma partícula, sendo $y_k(i)$ ora um gene ora uma componente de posição. Portanto, a aplicação sistemática de (4.3) resulta na decodificação de um cromossomo ou da posição de uma partícula \mathbf{Y} para \mathbf{X} , que descreve completamente uma FBG. Obviamente \mathbf{Y} pode ser constituído por um número de subvetores menor do que 5, caso um ou mais parâmetros sejam fixos. Por exemplo, para uma FBG de 1cm e de 100 seções seria possível deixar a heurística procurar apenas pela combinação mais adequada para os valores de ν_i fazendo $\mathbf{Y} = y_2$ e conservando $\delta n_{effi} = 1E-7$, $\Lambda_i = 1.55/(2n_{eff})\mu\text{m}$, $\Delta z_i = 100\mu\text{m}$ e $\phi_i = 0$. Neste caso, para $x_2^{min} = 0$ e $x_2^{max} = 4000$, a amplitude máxima da perturbação do índice de refração ficaria entre 0 e 4×10^{-4} .

4.2.2 Decodificação com limites variáveis ao longo da FBG

Como mostrado no Capítulo 2, perdas por inserção e lóbulos laterais pronunciados no espectro das FBGs podem ser reduzidos através da apodização, cujos formatos procuram minimizar gradualmente o produto $\nu_i \times \delta n_{effi}$ nas extremidades da grade. Filtros de alta qualidade empregados em sistemas de comunicação de alto desempenho podem adotar apodizações de formatos “fusiforme”, como perfis gaussiano ou cosseno-levantado. Mesmo em problemas de síntese que necessitem de perfis de apodização mais elaborados é certa a necessidade do caimento do produto $\nu_i \times \delta n_{effi}$ nas extremidades da FBG.

Portanto, é inútil permitir que a heurística realize testes com soluções candidatas cujos perfis são provavelmente inadequados. O próprio espaço de soluções pode ser reduzido se forem consideradas apenas o conjunto dos perfis que apresentem um caimento nas suas extremidades. Quando menor o espaço de soluções, mais eficiente poderá ser a heurística.

Para efetivar a restrição do espaço de soluções, a equação (4.3) pode ser adaptada como se segue:

$$\mathbf{x}_k(i) = \mathbf{x}_k^{min}(i) + (\mathbf{x}_k^{max}(i) - \mathbf{x}_k^{min}(i)) \times \mathbf{y}_k(i), \quad (4.4)$$

ou seja, os limites \mathbf{x}_k^{min} e \mathbf{x}_k^{max} passam a ser curvas (ou vetores) em função do número da seção i , onde $k = 1$ e/ou $k = 2$, pois δn_{effi} e ν_i ($k = 1$ e $k = 2$) definem a amplitude de perturbação do índice de refração.

Um caso particular da codificação empregando limites variáveis (como curvas) permite obter todo o vetor x_k de apenas um gene, isto é, a partir do escalar $y_k(1)$:

$$\mathbf{x}_k(i) = \mathbf{x}_k^{min}(i) + (\mathbf{x}_k^{max}(i) - \mathbf{x}_k^{min}(i)) \times \mathbf{y}_k(1). \quad (4.5)$$

4.2.2.1 Decodificação baseada em uma família de curvas - DFC

Dentre todos os 5 tipos de parâmetros do problema de síntese de FBGs, não raro para um destes reconhece-se que um tipo específico de formato da curva de apodização seria mais adequada ou viável. Por exemplo, a apodização cosseno-levantado para ν_i ($k = 2$). É possível deixar com que a heurística determine uma curva em particular dentre uma família de curvas entre $x_k^{min}(i)$ e $x_k^{max}(i)$ através de (4.4) com apenas um gene. Assim, a heurística pode se beneficiar de uma drástica redução da quantidade de dimensões do problema.

A Fig. 4.1 mostra três exemplos de decodificação para uma FBG de $M = 100$ utilizando: (a) a expressão (4.3) com $\mathbf{x}_k^{min} = 0$ e $\mathbf{x}_k^{max} = 4000$ aplicada sobre um vetor y_k aleatório; (b) a expressão (4.4) tendo $\mathbf{x}_k^{min}(i)$ e $\mathbf{x}_k^{max}(i)$ como curvas cosseno-levantado de amplitudes 1000 e 4000 respectivamente, aplicada sobre o mesmo vetor \mathbf{y}_k aleatório utilizado em “(a)”; e a expressão (4.5), com as mesmas curvas $\mathbf{x}_k^{min}(i)$ e $\mathbf{x}_k^{max}(i)$ empregadas em “(b)”. Considere que para este exemplo $\delta n_{effi} = 1 \times 10^{-7}$, o que implica na máxima amplitude de perturbação de índice igual a 4×10^{-4} .

A Fig. 4.2 exhibe algumas curvas intermediárias da família definida entre a reta $x_k^{min}(i)$ e a curva cosseno-levantado $\mathbf{x}_k^{max}(i)$. Diversos valores de $y_k(1)$ foram utilizados em (4.5). Este exemplo demonstra a possibilidade, por parte da heurística, de determinar o melhor formato

entre duas apodizações completamente diferentes.

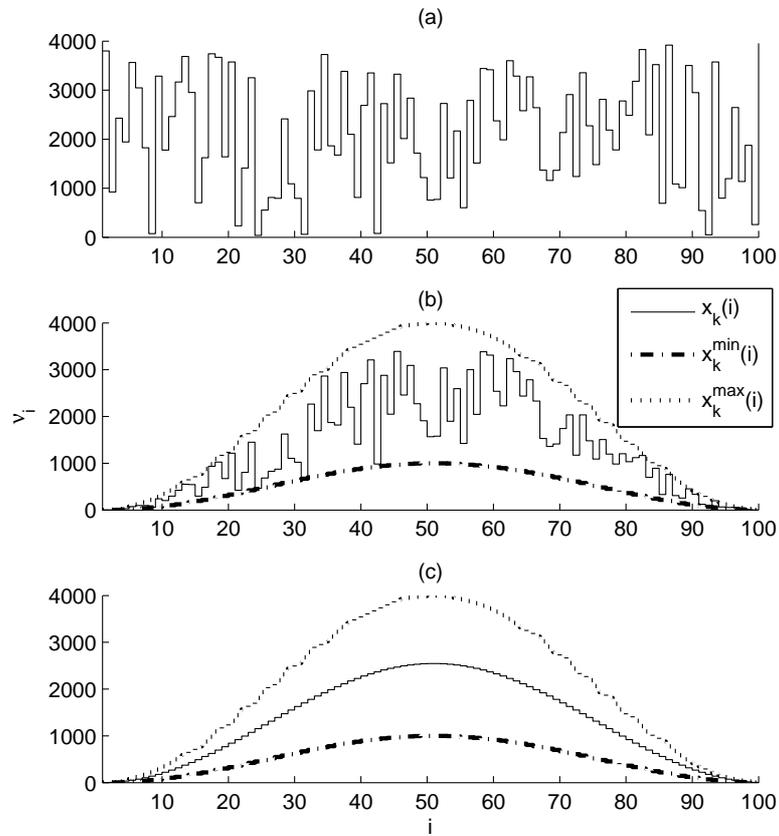


Figura 4.1: Exemplo onde três tipos de decodificação são comparadas: (a) empregando limites estáticos; (b) empregando curvas como limites e (c) empregando curvas como limites e apenas um escalar como parâmetro.

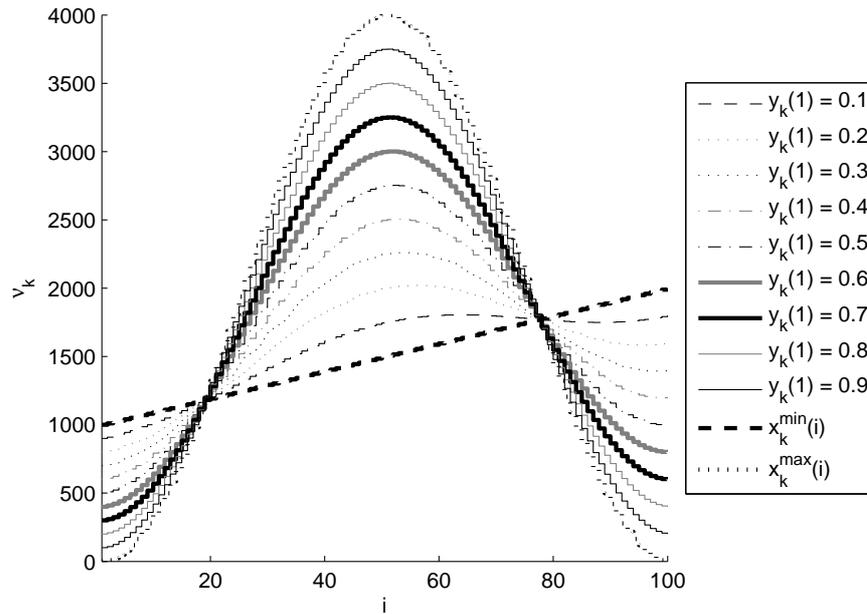


Figura 4.2: Exemplo onde várias curvas intermediárias, para diversos valores de $y_k(1)$, são mostradas entre as curvas limites $x_k^{\min}(i)$ e $x_k^{\max}(i)$.

4.2.3 Decodificação empregando suavização

Uma apodização “fusiforme” permite reduzir perdas de inserção bem como um possível efeito de cavidade da FBG (este último efeito é bastante característico nos espectros das grades uniformes). Porém, contrastes na perturbação do índice entre seções uniformes no interior da grade também podem ter efeitos igualmente indesejáveis. De fato, as apodizações normalmente empregadas nos projetos de FBGs são, em toda a sua extensão, curvas simples e suaves. Formas intrincadas e descontínuas além de possibilitar a formação de cavidades também podem ser difíceis de se fabricar. Sem qualquer tipo de restrição ou procedimento especial, heurísticas como o GA ou o PSO podem desperdiçar seus recursos pesquisando soluções candidatas indistintamente, avaliando também estas grades seguramente inadequadas.

Para eliminar do espaço de soluções as grades indesejáveis e reduzir consideravelmente o próprio espaço de soluções do problema, é preciso recorrer a procedimentos de decodificação especiais. Dois procedimentos são propostos: o primeiro baseado na suavização artificial restritiva; o segundo baseado na aplicação da “média em movimento”. Independentemente do tipo de técnica de decodificação suavizada, o vetor \mathbf{x}_k é dado em função de \mathbf{y}_k do modo como se segue:

$$\mathbf{x}_k = \text{suav}(\mathbf{y}_k). \quad (4.6)$$

Onde a função *suav* representa algum tipo de transformação sobre \mathbf{y}_k que resulte em um vetor \mathbf{x}_k “suavizado” de acordo com certas restrições e condições.

4.2.3.1 Decodificação suavizada restritiva - DSR

A suavização restritiva define *suav* em (4.6) através da seguinte relação:

$$\mathbf{x}_k(i) = \begin{cases} \mathbf{x}_k^{\min}(i) + (\mathbf{x}_k^{\max}(i) - \mathbf{x}_k^{\min}(i))\mathbf{y}_k(i) & \text{se } i = 1 \\ \max(\mathbf{x}_k^{\min}, \min(\mathbf{x}_k^{\max}, \mathbf{x}_k(i-1) + p * (\mathbf{x}_k^{\max}(i) - \mathbf{x}_k^{\min}(i))(2 * \mathbf{y}_k(i) - 1))) & \text{se } i \neq 1 \end{cases} \quad (4.7)$$

Onde o subscrito i representa o número da seção; $0 \leq \mathbf{y}_k(i) \leq 1$; $\mathbf{x}_{k \min}(i)$ e $\mathbf{x}_{k \max}(i)$ representam respectivamente o valor mínimo e máximo possíveis para $\mathbf{x}_k(i)$ tal que $\mathbf{x}_{k \min}(i) \leq \mathbf{x}_k(i) \leq \mathbf{x}_{k \max}(i)$; p , $|p| < 1$, é um parâmetro restritivo que limita a diferença entre $\mathbf{x}_k(i-1)$ e $\mathbf{x}_k(i)$; Finalmente, $\min()$ e $\max()$ são funções que respectivamente retornam o menor e o maior argumento dentre aqueles que lhes são passados.

A equação (4.7) define que o elemento $\mathbf{x}_k(i)$ para $i \neq 1$ será dado em função do elemento da seção anterior. A diferença entre ambos será limitada de modo que $|\mathbf{x}_k(i) - \mathbf{x}_k(i-1)| \leq p(|\mathbf{x}_k^{\max}(i) - \mathbf{x}_k^{\min}(i)|)$, o que na prática reduz severamente o espaço de buscas simplificando os esforços da heurística, muito embora não altere a dimensionalidade do problema.

4.2.3.2 Decodificação suavizada pela média em movimento - DSMM

A suavização (*smoothing*) é uma técnica bastante utilizada em processamento de imagens, estatística e filtragem digital. Dentre os diferentes algoritmos disponíveis, o mais comum é o baseado na *média em movimento* (ou média móvel). Tal como seu nome sugere, a média em movimento pode ser efetuada através de uma simples média aritmética não ponderada dos últimos eventos, tal como a média aritmética das últimas ou das mais próximas amostras do cromossomo ou da posição da partícula \mathbf{Y} , que passa a ser interpretado como uma seqüência temporal. Baseado neste conceito, uma forma simples de definir o procedimento *suav* seria através da seguinte expressão:

$$\mathbf{x}_k(i) = x_k^{\min}(i) + (x_k^{\max}(i) - x_k^{\min}(i)) \times \frac{1}{2 \Delta m + 1} \sum_{m=i-\Delta m}^{i+\Delta m} y_k(m) \quad (4.8)$$

Onde $\Delta m = \min(i - 1, \text{span}, M - i)$ e span representa o número de amostras tomadas à esquerda e à direita de i (além da própria amostra i) para o cálculo da média. Assim, para $\text{span} = 5$, a média entre 11 amostras contribuirá para cada posição de \mathbf{x}_k , desde que $i - 1$ ou $M - i$ não sejam inferiores a span , pois nesse caso um número máximo de pares de amostras simétricas (à esquerda e à direita) serão consideradas na média em movimento. Para $i = 1$ e $i = M$ nenhuma média será realizada e (4.8) retornará o mesmo que (4.4).

A DSMM é caracterizada pela simplicidade, o que a torna uma alternativa bastante atraente em relação à DSR. Embora não possa ser regulada de forma tão rigorosa como a DSR, que possui um parâmetro específico p , a suavização *smoothing* pode ser intensificada à medida que mais amostras vizinhas sejam consideradas no cálculo da média (parâmetro span).

A Fig. 4.3 compara apodizações para ν_i hipotéticas obtidas a partir de um cromossomo (ou partícula) $\mathbf{Y} = \mathbf{y}_2$ de 100 posições. A figura considera uma curva limite mínima constante $\mathbf{x}_k^{\min} = 0$ e uma máxima \mathbf{x}_k^{\max} cosseno-levantado de valor de pico igual a 4000. Este valor corresponde a uma amplitude de perturbação máxima $\nu_i \times \delta n_{eff i} = 4 \times 10^{-4}$ para $\delta n_{eff i} = 10^{-7}$. Das quatro curvas mostradas, a primeira foi obtida através de (4.4) sem qualquer suavização; a segunda foi obtida usando DSR com $p = 0.15$; a terceira DSMM com $\text{span} = 5$. A última curva é o limite máximo cosseno-levantado. Percebe-se que a decodificação restritiva não gera uma apodização que siga a original (não suavizada) de forma óbvia. Também satura-se (como observado no limite inferior), algo que a decodificação com média em movimento evita completamente. Em compensação, a decodificação restritiva é capaz de alcançar amplitudes máximas mais próximas do limite, algo bastante difícil para a decodificação com média em movimento.

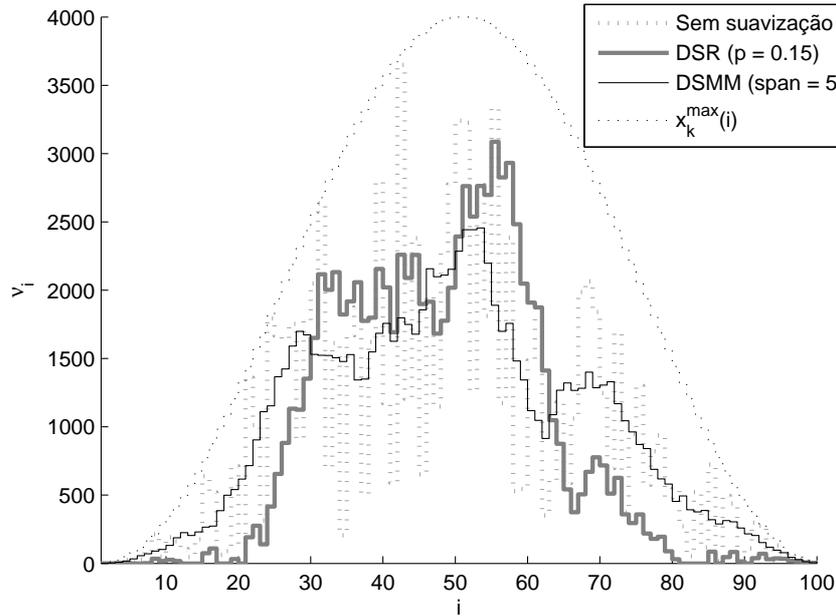


Figura 4.3: Comparação entre apodizações conseguidas para o mesmo cromossomo hipotético (ou equivalente PSO) utilizando a decodificação normal limitada por \mathbf{x}_k^{max} , suavização restritiva e suavização por média em movimento.

4.2.4 Decodificação usando interpolação

Para reduzir a dimensionalidade do problema e obter curvas de apodização suaves, propõe-se o uso de interpolação. Para propósito geral, considere o subvetor \mathbf{x}_k , onde $k = \{1, 2, 3, 4, 5\}$, como função de outro subvetor \mathbf{y}_k e de i :

$$\mathbf{x}_k(i) = \text{interp}(\mathbf{u}_k, i). \quad (4.9)$$

Onde i é o número da seção uniforme. A função *interp* representa o procedimento de interpolação, o qual considera que: (1) todas as amostras de \mathbf{x}_k e \mathbf{u}_k são homogeneamente espaçadas; (2) $\mathbf{x}_k(1) = \mathbf{y}_k(1)$ e $\mathbf{x}_k(M) = \mathbf{y}_k(M_k)$. Diferente da decodificação suavizada, o subvetor \mathbf{y}_k possui comprimento M_k ($M_k < M$) formando um cromossomo (ou equivalente PSO) \mathbf{Y} com um número de genes ou posições inferior da quantidade de parâmetros.

4.2.4.1 Decodificação usando interpolação linear - DIL

Neste tipo de interpolação a curva resultante é contemplada através da concatenação de inúmeros segmentos de reta. Cada ponto de \mathbf{x}_k é como pertencente a um destes segmentos, o qual pode ser definido como função de dois elementos de \mathbf{y}_k . A equação a seguir:

$$\mathbf{x}_k(i) = \begin{cases} x_k^{min}(i) + (x_k^{max}(i) - x_k^{min}(i))\mathbf{y}_k(1) & se \quad M_k = 1 \\ x_k^{min}(i) + (x_k^{max}(i) - x_k^{min}(i))((1 - q)\mathbf{y}_k(h) + q\mathbf{y}_k(h + 1)) & se \quad 1 < h < M_k \end{cases} \quad (4.10)$$

substitui (4.9), onde $q = z - h$; $h = floor(z)$, sendo $floor()$ uma função que retorna o número inteiro mais próximo, porém menor que o argumento; o parâmetro z é definido em função do número da seção i :

$$z = 1 + (M_k - 1) \times \frac{i - 1}{M - 1} \quad (4.11)$$

4.2.4.2 Decodificação usando spline quadrática - DSQ

Uma spline quadrática pode ser obtida a partir do vetor \mathbf{y}_k interpretado como um conjunto de pontos de controle. A curva resultante não passará em todos os pontos associados a \mathbf{y}_k senão no primeiro e no último ponto, ratificando este procedimento como uma espécie de ajuste de curva e não verdadeiramente uma interpolação. A equação (4.9), para este caso, pode ser redefinida como

$$\mathbf{x}_k(i) = \begin{cases} x_k^{min}(i) + (x_k^{max}(i) - x_k^{min}(i))[(1 - q)\mathbf{y}_k(h) + qB] & se \quad h = 1 \\ x_k^{min}(i) + (x_k^{max}(i) - x_k^{min}(i))[(1 - q)^2A + 2q(1 - q)\mathbf{y}_k(h) + q^2B] & se \quad 1 < h < M_k \\ x_k^{min}(i) + (x_k^{max}(i) - x_k^{min}(i))[(1 - q)A + q\mathbf{y}_k(h)] & se \quad h = M_k \end{cases} \quad (4.12)$$

onde $A = (\mathbf{y}_k(h - 1) + \mathbf{y}_k(h))/2$ e $B = (\mathbf{y}_k(h + 1) + \mathbf{y}_k(h))/2$. Neste caso o parâmetro h é definido em função de z como $h = round(z)$, onde $round()$ representa uma função de arredondamento que retorna o número inteiro mais próximo do argumento. O parâmetro z é definido através da mesma equação (4.11). O parâmetro q pode ser calculado a partir de z e h :

$$q = \begin{cases} 2z - 2 & \text{se } h = 1 \\ z - h + 0.5 & \text{se } 1 < h < M_k \\ 2z - 2h + 1 & \text{se } h = M_k \end{cases} \quad (4.13)$$

O procedimento de ajuste para spline quadrática descreve como várias curvas de Bézier podem ser conectadas entre si obtendo um resultado razoavelmente contínuo. É evidente, através de (4.12), que os pontos de conexão entre as curvas de Bézier situam-se nos pontos médios entre cada segmento de reta entre os pontos relativos aos elementos de \mathbf{y}_k . Conseqüentemente a curva ajustada interceptará os pontos médios dos segmentos de reta, assumindo a inclinação destes segmentos nestes mesmos pontos. Este comportamento permite um curva resultante pelo menos uma vez derivável. Nas semi-retas terminais, para $h = 1$ ou $h = M_k$, as curvas de Bézier são comutadas de quadrática para linear sem detrimento da continuidade. Para $M_k = 2$ o procedimento de ajuste é idêntico à interpolação linear.

4.2.4.3 Decodificação usando spline cúbica - DSC

Uma spline cúbica é composta por inúmeras curvas de Bézier cúbicas. Para este caso, a equação (4.9) pode ser substituída por:

$$\mathbf{x}_k(i) = \begin{cases} x_k^{\min}(i) + (x_k^{\max}(i) - x_k^{\min}(i))\mathbf{y}_k(1) & \text{se } M_k = 1 \\ x_k^{\min}(i) + (x_k^{\max}(i) - x_k^{\min}(i))[(1 - q)^3 A + 3q(1 - q)B + q^3 C] & \text{se } M_k \neq 1 \end{cases} \quad (4.14)$$

Onde A , B , C e D são pontos de controle dados em função dos elementos de \mathbf{y}_k ; $h = \text{floor}(z)$, sendo z dado por (4.11) e $q = z - h$. Os pontos de controle A , B , C e D podem ser obtidos a partir das seguintes equações:

$$A = \begin{cases} \mathbf{y}_k(h) & \text{se } h = 1 \\ \frac{\mathbf{y}_k(h-1) + 4\mathbf{y}_k(h) + \mathbf{y}_k(h+1)}{6} & \text{se } h > 1 \end{cases} \quad (4.15)$$

$$B = \frac{2\mathbf{y}_k(h) + \mathbf{y}_k(h+1)}{3} \quad (4.16)$$

$$C = \frac{2\mathbf{y}_k(h+1) + \mathbf{y}_k(h)}{3} \quad (4.17)$$

$$D = \begin{cases} \mathbf{y}_k(h) & \text{se } h = M_k - 1 \\ \frac{\mathbf{y}_k(h) + 4\mathbf{y}_k(h+1) + \mathbf{y}_k(h+2)}{6} & \text{se } h < M_k - 1 \end{cases} \quad (4.18)$$

A Fig. 4.4 exibe apodizações obtidas com a decodificação interpolada linear e ajustada, spline quadrática e spline cúbica, a partir do mesmo cromossomo (ou equivalente PSO) aleatório hipotético. A Fig. 4.4 (a) exibe as apodizações obtidas para limites constante $\mathbf{x}_k^{min} = 0$ e $\mathbf{x}_k^{max} = 4000$ ($\delta n_{effi} = 10^{-7}$). A Fig. 4.4 (b) exibe as apodizações obtidas para o limite inferior $\mathbf{x}_k^{min} = 0$ e superior \mathbf{x}_k^{max} cosseno-levantado de amplitude igual a 4000. Percebe-se que a spline cúbica é ligeiramente mais suave do que a spline quadrática, embora as duas sejam muito parecidas. Foi empregado um cromossomo de apenas 10 valores para obter uma apodização com 100 seções, embora efetivamente apenas 8 genes tenham sido considerados na Fig. 4.4 (b), devido à curva limite máxima decair a zero nas extremidades.

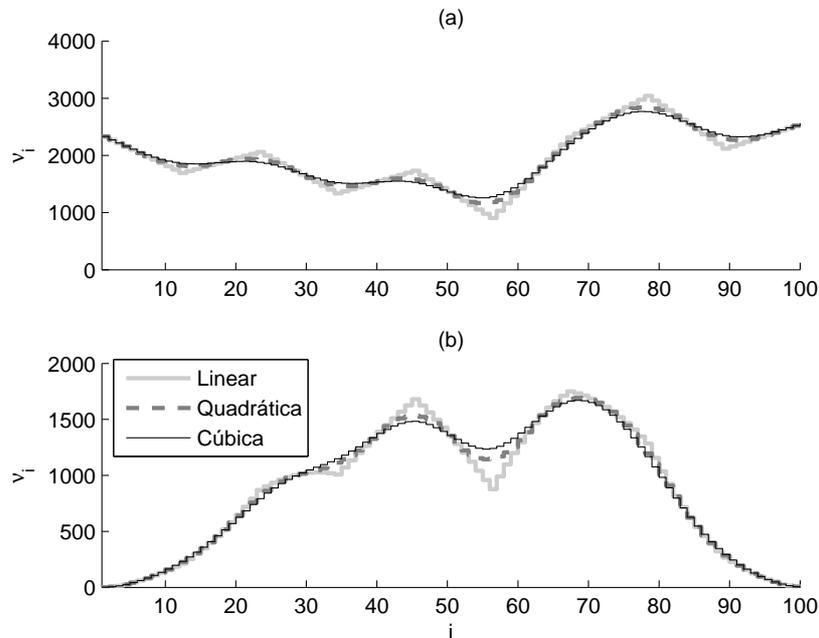


Figura 4.4: Comparação entre apodizações conseguidas para o mesmo cromossomo hipotético (ou equivalente PSO) utilizando a decodificação interpolada linear, ajustada quadrática e cúbica para limites constantes (a) e limites apodizados (b).

4.2.5 Deslocamentos de fase

Todas as técnicas de decodificação anteriormente mostradas podem ser aplicadas a todos os parâmetros que definem das seções uniformes do modelo escolhido para as FBGs, embora sejam extremamente inconvenientes para o parâmetro ϕ_i . Como mostrado no Capítulo

2, o deslocamento discreto na fase é uma singularidade rara nas FBGs, sendo empregadas de modo contido em projetos bastante particulares. A representação dada por (4.2), onde deslocamentos podem ocorrer entre quaisquer seções, apesar de correta, não é consistente com a realidade. Na prática, raramente mais do que um único deslocamento é intencionalmente provocado, obviamente exceto nas grades amostradas [16].

Portanto, a forma mais conveniente para a definição do vetor \mathbf{y}_5 seria reservar dois genes para cada deslocamento: o primeiro gene guarda a informação sobre a posição do deslocamento e o segundo a informação sobre o valor do deslocamento, ambos normalizados. Assim, $\mathbf{y}_5 = \{y_1^p, y_1^d, y_2^p, y_2^d, \dots, y_{M_5/2}^p, y_{M_5/2}^d\}$. O valor padrão para \mathbf{x}_5 deve ser zero, o que não é explícito a partir do subvetor \mathbf{y}_5 (que faz parte do cromossomo \mathbf{Y}). O processo de decodificação não pode ser descrito através de uma simples função como (4.3), (4.4), (4.5), (4.6), (4.7) e (4.8), mas por um algoritmo:

Parâmetros:

\mathbf{y}_5 : o cromossomo (ou equivalente PSO)

M_5 : comprimento de \mathbf{y}_5

\mathbf{X} : matriz que representa a FBG

M : número de seções de \mathbf{X}

Retorna:

\mathbf{x}_5 : subvetor de \mathbf{X} referente aos parâmetros ϕ_i

01. $L = 0$;

02. Para $i = 1$ até M :

03. $\mathbf{x}_5(i) = 0$;

04. $L = L + \mathbf{X}(4, i)$;

05. Para $m = 1$, enquanto $m \leq M_5$, fazendo $m = m + 2$:

06. $i = 1$;

07. $z = \mathbf{X}(4, i)$;

08. Enquanto $z/L < \mathbf{y}_5^{min}(m) + (\mathbf{y}_5^{max}(m) - \mathbf{y}_5^{min}(m))\mathbf{y}_5(m)$ e $i < M$:

09. $i = i + 1$;

10. $z = z + \mathbf{X}(4, i)$;

11. $\mathbf{x}_5(i) = \mathbf{y}_5^{min}(m + 1) + (\mathbf{y}_5^{max}(m + 1) - \mathbf{y}_5^{min}(m + 1))\mathbf{y}_5(m + 1)$;

12. retorna \mathbf{x}_5 ;

Figura 4.5: Função para a decodificação da subseção de cromossomo \mathbf{y}_5 .

As linhas 01 a 04 procuram calcular o comprimento total da FBG (armazenado em

L) e, ao mesmo tempo, iniciar o vetor \mathbf{x}_5 com valores nulos. É importante observar que a estrutura \mathbf{X} é manipulada como uma matriz, onde cada linha está associada a um respectivo parâmetro. A linha 4 possui as espessuras das seções.

O laço que se inicia na linha 05 realiza a decodificação de fato (incrementando m de dois enquanto $m \leq M_5$). No seu interior, o laço “enquanto”, que se inicia na linha 08, procura identificar a que seção cada deslocamento de fase pertence. É interessante notar que tanto as posições dos deslocamentos quanto eles próprios são limitados por \mathbf{y}_5^{min} e \mathbf{y}_5^{max} , vetores com o mesmo número de elementos que o subvetor (sub-cromossomo) \mathbf{y}_5 . Os limites para as posições de deslocamento são normalizadas, enquanto os valores podem ser dados em radianos (normalmente $\mathbf{y}_5^{min}(i) = -\pi$ e $\mathbf{y}_5^{max}(i) = +\pi \forall i = 2, 4, 6, \dots, M_5$). Valores limites em função do número da seção foram utilizados em (4.3), (4.4), (4.5), (4.6), (4.7) e (4.8). Porém, para ϕ_i parece ser mais conveniente associar os valores limites dos parâmetros da grade direta ou indiretamente ou aos genes através de \mathbf{y}_5^{min} e \mathbf{y}_5^{max} . Graças a estes valores limites, é possível fazer com que a heurística também procure, por exemplo, a melhor posição para um deslocamento de fase fixo, ou procure o melhor deslocamento para uma posição específica.

4.3 A função objetivo - FO

A aplicação de heurísticas para a síntese de FBGs tem como objetivo descobrir combinações ótimas para os parâmetros contidos na estrutura \mathbf{X} , para que o espectro de refletividade adquira um formato conveniente. Para tanto, a FO deverá retornar um valor proporcional à semelhança entre a curva do espectro desejada (alvo) e o espectro calculado através da formulação matricial apresentada no Capítulo 2.

A comparação entre curvas é obviamente baseada na amostragem. Geralmente o espectro de refletividade para uma solução candidata é calculada para um certo número de comprimentos de onda homogeneamente espaçados. Estas amostras de refletividade podem ser comparadas com os pontos relativos da curva alvo (obtidos por interpolação, se necessário) empregando o erro médio quadrático (*Root Mean Square* - RMS):

$$FO_1(\mathbf{X}) = \sqrt{\frac{1}{S} \sum_{n=1}^S w_n [r(\mathbf{X}, \lambda_n) - r_A(\lambda_n)]^2} \quad (4.19)$$

onde S representa o número total de amostras, w_n representa o peso associado à amostra n . A função $r()$ representa a técnica de análise matricial e retorna a refletividade para a grade \mathbf{X} e para o comprimento de onda λ_n . A função r_A representa a refletividade desejada para a

grade no comprimento de onda λ_n , o qual é dado por:

$$\lambda_n = \lambda_{min} + (\lambda_{max} - \lambda_{min}) \frac{n-1}{S-1} \quad (4.20)$$

onde λ_{min} e λ_{max} são os comprimentos de onda extremos que definem a janela espectral do problema.

Observa-se que a FO_1 dada por (4.19) deve ser minimizada. Seria possível transformar este problema de minimização em maximização ao definir a FO como $(FO_1(\mathbf{X}))^{-1}$ ou simplesmente $-(FO_1(\mathbf{X}))$. Também é oportuno observar que o termo $1/S$ pode ser removido de (4.19) provavelmente sem qualquer prejuízo à heurística, dado que esta apenas necessita que o valor retornado pela FO continue sendo proporcional à diferença entre as curvas de refletividade calculada e alvo. Em [53] de fato uma FO semelhante a (4.19) é empregada sem o fator $1/S$, sem a raiz quadrada e com pesos w_n seguindo uma gaussiana em função do comprimento de onda e alcançando valor máximo no centro da janela espectral. Em [7] o fator $\frac{1}{S}$ também é dispensado, porém a raiz quadrada é substituída por um valor fracionário em favor de uma expressão mais generalizada:

$$FO_2(\mathbf{X}) = \left\{ \sum_{n=1}^S w_n |r(\mathbf{X}, \lambda_n) - r_A(\lambda_n)|^\varpi \right\}^{\frac{1}{\varpi}} \quad (4.21)$$

onde o expoente ϖ tem o poder de determinar o quanto as maiores diferenças entre $r(\mathbf{X}, \lambda_n)$ e $r_A(\lambda_n)$ serão mais valorizadas em detrimento das menores. Para $\varpi = 1$, (4.21) representaria uma simples soma ponderada; para $\varpi = 2$, (4.21) representaria uma simples métrica euclidiana e para $\varpi \rightarrow \infty$ (4.21) seria equivalente a:

$$FO_3(\mathbf{X}) = \max \{w_1|r(\mathbf{X}, \lambda_1) - r_A(\lambda_1)|, w_2|r(\mathbf{X}, \lambda_2) - r_A(\lambda_2)|, \dots, w_S|r(\mathbf{X}, \lambda_S) - r_A(\lambda_S)|\} \quad (4.22)$$

onde a função $\max\{\}$ retorna o máximo valor dentre todos os argumentos.

Além da refletividade, outras grandezas podem ser importantes para a qualificação da grade. Módulos da dispersão D maiores do que zero no interior da banda de máxima refletividade é indesejável em muitos projetos de filtros ópticos. Neste caso, (4.22) pode ser adaptada como se segue:

$$FO_4(\mathbf{X}) = \max \{|D(\mathbf{X}, \lambda_1)|, |D(\mathbf{X}, \lambda_2)|, \dots, |D(\mathbf{X}, \lambda_n)|, |D(\mathbf{X}, \lambda_{n+1})|, \dots, |D(\mathbf{X}, \lambda_S)|\} \quad (4.23)$$

onde $D(\mathbf{X}, \lambda_n)$ retorna a dispersão para a grade \mathbf{X} para o comprimento de onda λ_n .

No exemplo clássico apresentado em [7] o interesse não é apenas em obter filtros refletores de alta seletividade, mas também garantir dispersão mínima no interior da banda de máxima reflexão. A alternativa empregada na referência foi de combinar linearmente FO_2 e FO_4 como $FO_5(\mathbf{X}) = f FO_2(\mathbf{X}) + (1 - f) FO_4(\mathbf{X})$, onde $0 \leq f \leq 1$. Este tipo de combinação entre funções objetivo é um artifício simples, mas que pode ser vantajosamente substituído pela otimização multiobjetivo. Neste caso, as funções objetivo são conservadas separadas, com múltiplos valores objetivos associados a cada indivíduo ou partícula e celebrados através de um esquema geralmente mais elaborado do que o empregado nas heurísticas mono-objetivo. Apesar disso, esta abordagem não será estudada neste trabalho por estar além do seu escopo; além disso, todos os desenvolvimentos mostrados neste capítulo independem do tipo de otimização, sendo aplicáveis tanto à otimização mono quanto a multiobjetivo.

A síntese de filtros ópticos objetivando a formatação do espectro do filtro sempre recorrerá a amostragens e chamadas sistemáticas à técnica de análise (para cada comprimento de onda). Considerando que as operações numéricas dos mecanismos próprios do PSO ou GA sejam menos ostensivas (computacionalmente) do que o conjunto de operações associado a uma chamada de, por exemplo, da função $r(\mathbf{X}, \lambda_n)$, o tempo computacional associado ao desenvolvimento da heurística deverá ser proporcional a S . Logo, a redução do número total de amostras deverá se refletir diretamente e proporcionalmente no desempenho computacional total do processo de otimização.

Entretanto, a simples redução da quantidade de amostras afeta a confiabilidade do valor objetivo², fazendo da comparação entre dois indivíduos (ou partículas) uma operação muito mais sujeita a erros. Em outras palavras, a seleção de indivíduos no GA ou o reconhecimento de uma partícula no PSO seria prejudicada, possivelmente impedindo o funcionamento apropriado destas heurísticas.

Portanto, existe um conflito entre a necessidade de desempenho computacional e a necessidade de eficiência da heurística, cujo equilíbrio é desejável e também muito difícil de se alcançar.

²Neste capítulo o termo “valor objetivo” refere-se ao resultado retornado pela FO. Substitui “aptidão”, que é associado à idéia de que a FO deveria sempre retornar um valor diretamente proporcional à qualidade da solução candidata. Pelo contrário, neste capítulo a FO assume a forma de um cálculo de erro, sempre retornando valores relativamente pequenos para soluções de alta qualidade.

4.3.1 Amostragem progressiva - AP

Embora não seja possível reduzir impunemente a quantidade total de amostras S empregadas nas funções objetivo, seria possível determinar o valor mínimo aceitável através de uma amostragem progressiva. Propõe-se um processo que se iniciaria com uma amostragem inicial uniforme muito pequena e certamente insuficiente. Então, novas amostras seriam sistematicamente inseridas nos comprimentos de onda médios entre as existentes até que algum critério de parada seja atingido. Se cada iteração do processo de fato inserir novas amostras na meia distância de todos os intervalos, então no total $S_{k-1} - 1$ seriam inseridas por iteração, onde o subscrito k representa o número da iteração corrente. Observa-se que todo custo computacional deverá concentrar-se quase completamente no processo de obtenção dos valores de refletividade para cada comprimento de onda; de posse destes valores, o custo associado à computação da FO pode ser desprezível.

Para FO_1 e FO_3 , o cálculo do erro entre os valores objetivos obtidos para as iterações k e $k - 1$ permitiria detectar a convergência da FO: quando o erro calculado fosse inferior ao mínimo adotado, o processo de amostragem progressiva seria interrompido. Para FO_2 , esse mesmo critério de parada não seria apropriado pela ausência do fator $1/S$, o que impede a convergência da FO quando $S \rightarrow \infty$. Um número máximo de iterações poderia constituir uma condição de parada mais apropriada neste caso.

A Fig. 4.6 exhibe o algoritmo básico para a amostragem progressiva. A FO adotada para este exemplo é baseada em (4.19). As linhas iniciais, de 01 a 04, realizam a amostragem inicial e determinam o valor objetivo inicial. As linhas 05 a 15 realizam a amostragem progressiva, onde as atribuições das linhas 07 e 08 determinam o redimensionamento da matriz \mathbf{M} (pelas atribuições a posições inexistentes). O operador “retorna” na linha 13 tem o poder para interromper o algoritmo. Observa-se que a curva alvo é armazenada na matriz **alvo** que organiza os comprimentos de onda na primeira coluna, os valores alvo de refletividade na segunda coluna e possui S_A linhas. A linha 10 determina a ordenação da matriz \mathbf{M} em função dos comprimentos de onda armazenados na primeira coluna. Como geralmente $S_A \neq S$, recorre-se a um procedimento de interpolação linear representado pela função *linterp()* nas linhas 04 e 11.

A Fig. 4.7 exhibe as curvas de valor calculado para a FO em função do número de amostra para três grades diferentes, utilizando o algoritmo da Fig. 4.6 adaptado para FO_1 em (a), para FO_2 com $\varpi = 2$ em (b) e para FO_3 em (c). Os critérios de parada foram $k_{max} = 8$ e $erro = 0.025$. O número de amostras iniciais foram 3. A curva alvo escolhida conserva-se em zero exceto no interior do intervalo de comprimentos de onda $[1.5498, 1.5502]$ em micrômetros.

A Fig. 4.8 exhibe as apodizações de ν_i para FBG1, FBG2 e FBG3 (a) e as respectivas

Parâmetros:

\mathbf{X} : estrutura que representa uma FBG

S : o número de amostras iniciais

λ_{min} : comprimento de onda mínimo

λ_{max} : comprimento de onda máximo

alvo : matriz $S_A \times 2$ representando a curva alvo

erro : máximo erro admissível para o cálculo do valor objetivo

k_{max} : número máximo de iterações

Retorna:

f : valor objetivo

01. Para $n = 1$ até S :

02. $\mathbf{M}[n, 1] = \lambda_{min} + (\lambda_{max} - \lambda_{min})(n - 1)/(S - 1)$;

03. $\mathbf{M}[n, 2] = r(\mathbf{X}, \mathbf{M}[n, 1])$;

04. $f_0 = f = \sqrt{(1/S) \sum_{n=1}^S |\mathbf{M}[n, 2] - \text{linterp}(\mathbf{alvo}, \mathbf{M}[n, 1])|^2}$;

05. Para cada iteração k , de $k = 1$ até k_{max} :

06. Para $n = 1$ até $S - 1$:

07. $\mathbf{M}[S + n, 1] = (\mathbf{M}[n, 1] + \mathbf{M}[n + 1, 1])/2$;

08. $\mathbf{M}[S + n, 2] = r(\mathbf{X}, \mathbf{M}[S + n, 1])$;

09. $S = S + (S - 1)$;

10. Ordene a matriz \mathbf{M} em função de sua primeira coluna;

11. $f = \sqrt{(1/S) \sum_{n=1}^S |\mathbf{M}[n, 2] - \text{linterp}(\mathbf{alvo}, \mathbf{M}[n, 1])|^2}$;

12. Se $\text{erro} \geq |f - f_0|/|f|$:

13. retorna f ;

14. $f_0 = f$;

15. retorna f ;

Figura 4.6: Função de amostragem progressiva.

curvas de refletividade (b). As apodizações foram concebidas para $\delta n_{eff i} = 10^{-12}$ de modo que $\delta n_{eff 2} \times \nu_i$ alcance o valor de no máximo 5×10^{-4} . Particularmente *FBG2* foi feita a partir do perfil cosseno-levantado de *FBG1* através da soma de um ruído branco com os valores das amostras maiores ou iguais a zero e menores ou iguais a 10^8 . A apodização para a grade uniforme *FBG3* foi definida como a média dos valores ν_i da grade *FBG2*. Todas as três grades utilizaram seções de $\Delta z_i = 100 \mu\text{m}$ (comprimento total de 1cm), $\Lambda_i = 1.55/(2n_{eff})$ ($n_{eff} = 1.45$) e (obviamente) $\phi_i = 0$. É interessante notar que *FBG2* parece ser visualmente

superior a FBG3, porém os valores de FO calculados durante a AP afrontam esta idéia para FO_1 e FO_2 (Fig. 4.7 (a) e (b)). Outro aspecto interessante é a convergência para a AP utilizando FO_3 (Fig. 4.7 (c)) apenas para uma quantidade de amostras superior a 100.

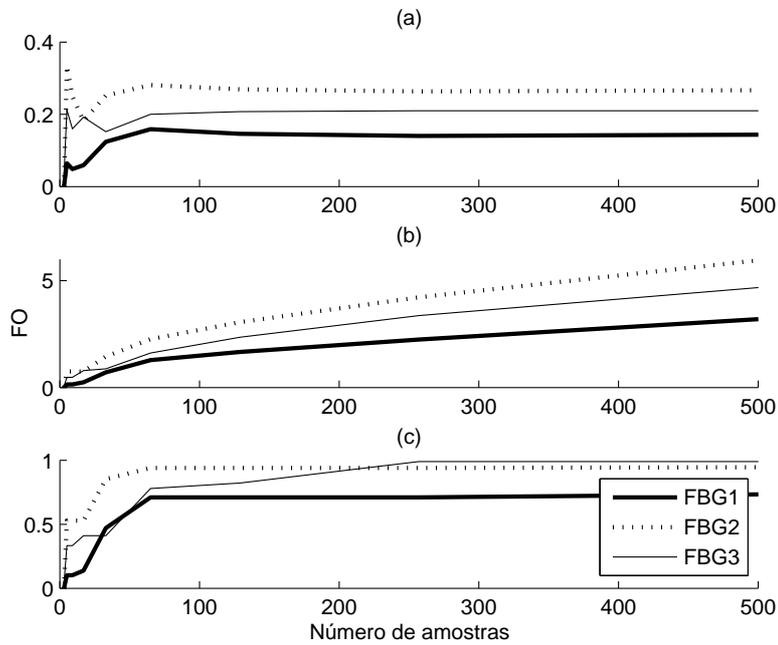


Figura 4.7: Curvas do valor objetivo em função do número de amostras para três tipos de FO diferentes: (a) para o erro RMS (FO_1), (b) para o erro quadrático cumulativo (FO_2) e (c) para o erro absoluto máximo (FO_3).

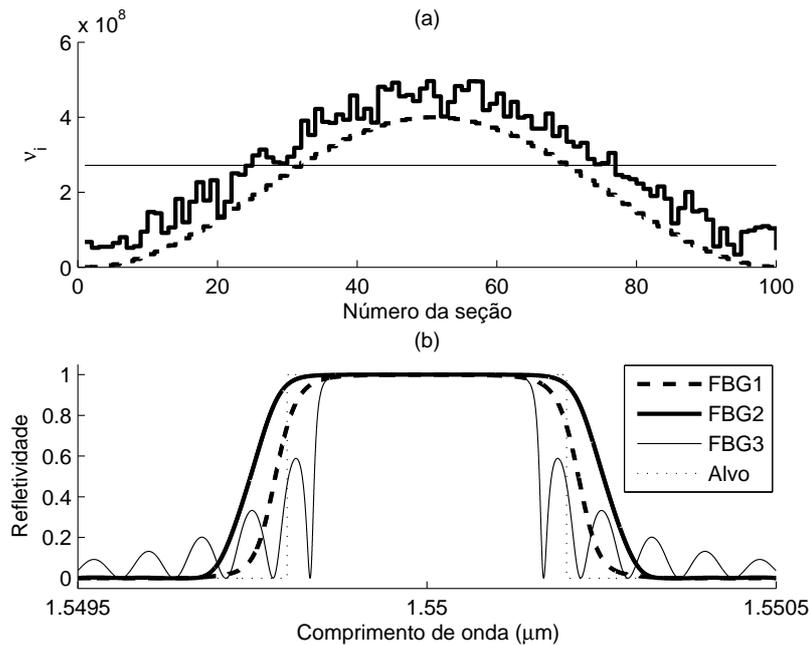


Figura 4.8: Três FBGs diferentes exemplo: (a) curvas de apodização para ν_i ; (b) curvas de refletividade.

A relação entre as heurísticas e a função objetivo pode revelar um outro critério mais interessante para a amostragem progressiva: a diferenciação entre soluções candidatas. Os valores objetivos são importantes para distinguir soluções candidatas, isto é, se uma solução é ou não é superior a outra. Porém, “o quanto uma solução candidata é melhor do que outra” é uma informação que pode ser irrelevante, por exemplo, na aplicação do GA com a seleção por torneio. Portanto, a amostragem progressiva poderia ser realizada até o momento onde o erro entre valores objetivos de iterações consecutivas seja suficientemente pequeno a ponto dos *ranks*³ das soluções candidatas poderem ser considerados estáveis.

4.3.2 Amostragem adaptativa - AA

As expressões para a FO até o momento foram suportadas por (4.20), que estabelece todos os comprimentos de onda homogeneamente espaçados. Entretanto, os espaçamentos poderiam ser inversamente proporcionais, por exemplo, à derivada da curva de refletividade, o que implicaria em mais amostras nas regiões mais conturbadas do espectro de refletividade e menos amostras nas regiões aplanadas. Esta estratégia poderia valorizar as particularidades de

³Aqui, entende-se por *rank* a ordem ou posição que solução candidata ocupa dentro de uma população ou de um exame representado por uma lista ordenada em função do valor objetivo.

cada espectro de refletividade e melhorar a diferenciação entre soluções candidatas da mesma população ou do mesmo enxame.

Parâmetros:

\mathbf{X} : estrutura que representa uma FBG

S : o número de amostras iniciais

λ_{min} : comprimento de onda mínimo

λ_{max} : comprimento de onda máximo

alvo : matriz $S_{A \times 2}$ representando a curva alvo

Δr : máximo diferença admissível para o cálculo do valor objetivo

Retorna:

f : valor objetivo

01. Para $n = 1$ até S :
 02. $\mathbf{M}[n, 1] = \lambda_{min} + (\lambda_{max} - \lambda_{min})(n - 1)/(S - 1)$;
 03. $\mathbf{M}[n, 2] = r(\mathbf{X}, \mathbf{M}[n, 1])$;
 04. Faça:
 05. $n_{ins} = 0$;
 06. Para $n = 1$ até $S - 1$:
 07. Se $|\mathbf{M}[n, 2] - \mathbf{M}[n + 1, 2]| > \Delta r$:
 08. $\mathbf{M}[S + n, 1] = (\mathbf{M}[n, 1] + \mathbf{M}[n + 1, 1])/2$;
 09. $\mathbf{M}[S + n, 2] = r(\mathbf{X}, \mathbf{M}[S + n, 1])$;
 10. $n_{ins} = n_{ins} + 1$;
 11. $S = S + n_{ins}$;
 12. Ordene a matriz \mathbf{M} em função de sua primeira coluna;
 13. Enquanto $n_{ins} > 0$;
 14. $f = \sqrt{(1/S) \sum_{n=1}^S |\mathbf{M}[n, 2] - \text{interp}(\mathbf{alvo}, \mathbf{M}[n, 1])|^2}$;
 15. retorna f ;
-

Figura 4.9: Função de amostragem adaptativa.

Propõe-se o processo de amostragem adaptativa. Partindo de uma fraca amostragem inicial homogênea, cada segmento da curva de refletividade entre amostras consecutivas é avaliado. Se a diferença entre valores de refletividade for superior a um mínimo adotado, uma nova amostra será inserida no comprimento de onda médio do segmento. O processo se repete até que mais nenhuma nova amostra seja inserida. A Fig. 4.9 exhibe este algoritmo. A amostragem inicial é realizada nas linhas iniciais de 01 a 04. O laço das linhas 05 a 13

realizam a amostragem adaptativa adicionando condicionalmente uma nova amostra a cada novo intervalo. A condição de permanência no interior do laço principal é averiguada na linha 13. O valor objetivo é computado apenas no final do algoritmo na linha 14.

A Fig. 4.10 exhibe a evolução do valor calculado da função objetivo usando AA em função da quantidade de amostras, substituindo a expressão da linha 14 do algoritmo (Fig. 4.9) adequadamente por equivalentes a (a) FO_1 , (b) FO_2 ($\varpi = 2$) e (c) FO_3 . Embora o algoritmo não calcule f a não ser no final, uma adaptação especial permitiu monitorar o comportamento desta variável ao longo das iterações. As três FBGs utilizadas foram as mesmas empregadas para os exemplos anteriores de AP. A amostragem inicial homogênea foi feita com 3 amostras. O valor para o parâmetro de controle Δr foi de 0.02, número considerado suficiente, abaixo do qual acusa-se sempre que FBG1 melhor que FBG3 melhor que FBG2 para FO_1 , FBG1 melhor que FBG2 melhor que FBG3 para FO_1 e FO_3 . Algo bastante interessante a cerca da amostragem adaptativa é a sua relevante capacidade de economia de amostras. Como esperado, quanto mais ‘difícil’ o espectro de refletividade (repleto de máximos, mínimos e lóbulos) mais amostras são necessárias, chegando a 145 e 163 respectivamente para FBG1 e FBG2, e 473 para FBG3.

O comportamento conflitante entre Fig. 4.10 (a) e (b) explica-se pela proporcionalidade do valor objetivo em função da raiz quadrada da quantidade de amostras apenas para FO_2 : como o espectro de FBG3 demanda mais que o dobro de amostras que FBG1 e FBG2, o valor objetivo acaba por ultrapassar os valores objetivos de FBG1 e FBG2, corretamente determinando FBG3 como inferior às outras duas grades. Para FO_3 o comportamento da amostragem adaptativa permite obter uma correta convergência com pouco mais do que 100 amostras, quase a metade das amostras necessárias pela AP. De fato, é previsível que a AA ofereça resultados bastante confiáveis para qualquer FO baseada em FO_3 devido a presença de mais pontos nas imediações de todas as importantes perturbações do espectro de refletividade. Uma outra observação oportuna seria sobre a rápida convergência de FO_3 para FBG3, porém a amostragem adaptativa continua desnecessariamente até atingir 473 amostras. Portanto, seria interessante agregar o erro numérico como mais um critério de parada do mesmo modo como feito para a AP.

A Fig. 4.11 exhibe os espectros de refletividade para as grades FBG1, FBG2 e FBG3 obtidas através da AA. Observa-se que apenas uma amostra é colocada no centro da banda de máxima refletividade devido a amostragem inicial com apenas 3 amostras. Em compensação, as regiões mais críticas próximo da borda da banda de máxima reflexão e lóbulos laterais são repletas.

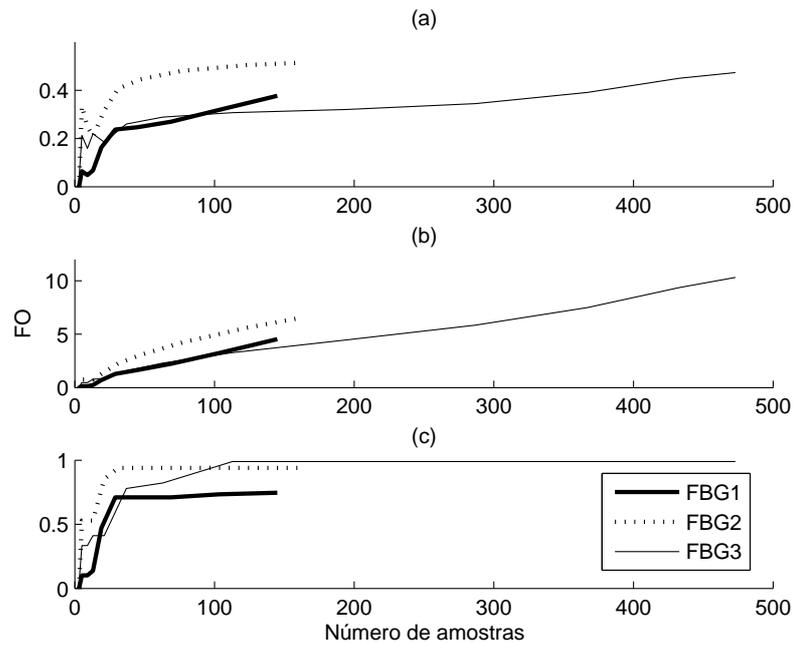


Figura 4.10: Curvas do valor objetivo em função do número de amostras para três tipos de FO diferentes: (a) para o erro RMS (FO_1), (b) para o erro quadrático cumulativo (FO_2) e (c) para o erro absoluto máximo (FO_3).

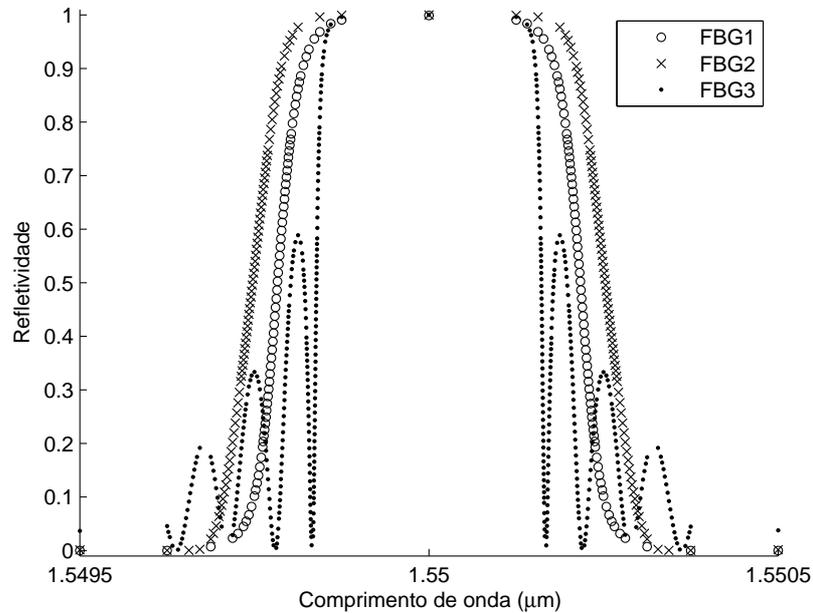


Figura 4.11: Curvas de refletividade para FBG1, FBG2 e FBG3 obtidas através da amostragem adaptativa.

4.3.3 Amostragem dinâmica - AD

A diversidade nos GA e PSO varia dinamicamente em função das iterações, em média, sempre decaindo até o final da otimização. A diversidade revela o quão diferentes são os indivíduos da população ou do enxame. No início da otimização, a diferença entre os indivíduos é geralmente muito mais acentuada do que no final. Assim, poucas amostras são necessárias para distinguir profundas diferenças entre soluções candidatas e, obviamente, muitas amostras são necessárias quando as soluções candidatas tornam-se substancialmente parecidas no final da otimização. Propõe-se, portanto, uma amostragem dinâmica onde o parâmetro S de FO_1 , FO_2 ou FO_3 seria dado em função do número de gerações ou iterações, por exemplo, de forma linear:

$$S_g = S_{min} + (S_{max} - S_{min}) \frac{g - 1}{G - 1} \quad (4.24)$$

Onde S_g representa o número de amostras a ser utilizado na geração g ; S_{min} e S_{max} são, respectivamente, o mínimo e o máximo valor possível para S_g ; G representa o número máximo de gerações (ou iterações).

Comparativamente à amostragem progressiva e adaptativa, esta abordagem tem a vantagem de ser extremamente simples. A desvantagem seria na determinação dos parâmetros S_{min} e S_{max} , que são dependentes de G . Uma alternativa para tornar o processo mais “inteligente” seria calcular o número de amostras S_g em função da diversidade da população na geração e não em função g .

4.3.4 Amostragem aleatória - AAL

Na amostragem com amostras determinísticas a FO sempre retornará o mesmo valor para a mesma solução candidata. A medida que a quantidade de amostras é reduzida, o valor objetivo pode passar de completamente correto a incorreto de forma abrupta. Na maioria dos casos, a redução arbitrária do número de amostras é sempre prejudicial à heurística.

Entretanto, o GA e o PSO têm considerável tolerância a uma FO probabilística. Neste tipo de FO, à medida que a quantidade de amostras é reduzida, a variância do valor objetivo aumenta. A diferenciação entre soluções candidatas é prejudicada, mas a probabilidade de erro em comparações entre duas soluções pode conservar-se favorável para uma quantidade de amostra bastante reduzida. Como a heurística freqüentemente conserva a mesma solução candidata ou soluções similares por várias iterações, os algoritmos tendem a retificar comparações eventualmente mal sucedidas.

A amostragem aleatória pode ser realizada através de uma simples modificação de 4.20:

$$\lambda_n = \lambda_{min} + (\lambda_{max} - \lambda_{min}) \frac{n - 1 + u_n}{S} \quad (4.25)$$

onde $u_n = u(0, 1)$, isto é, a realização de uma variável aleatória uniforme entre 0 e 1. É suficiente, entretanto, fazer $u_1 = u_2 = \dots u_n = u(0, 1)$, onde todas os comprimentos de onda se conservariam equidistantes, porém aleatoriamente deslocados em relação a λ_{min} . A vantagem desta segunda abordagem seria computacional, pois menos números aleatórios precisariam ser gerados. De fato, alguns geradores de números pseudo aleatórios mais sofisticados podem ser computacionalmente ostensivos [52].

Por exemplo, para FBG1 e FBG2 usando FO_1 com apenas 3 comprimentos de onda homoganeamente espaçados e aleatoriamente posicionados dados por (4.25), a probabilidade de sucesso estimada para a comparação entre estas duas grades hipotéticas seria de aproximadamente 610 por 1000 tentativas.

4.3.4.1 Amostragem aleatória aplicada ao GA: o GA acelerado

A redução da quantidade de amostras pode oferecer um considerável ganho computacional, uma vez que o tempo de processamento de uma chamada da FO é diretamente proporcional à quantidade de amostras.

Entretanto, embora a amostragem aleatória possa garantir o funcionamento GA, a confiabilidade das comparações entre soluções candidatas deverá interferir no procedimento de seleção, afetando a pressão seletiva e reduzindo a taxa de convergência do GA. Provavelmente mais gerações seriam necessárias para alcançar a mesma qualidade de solução final em comparação a um GA empregando uma FO com grande quantidade de amostras fixas uniformemente espaçadas. Felizmente, o elitismo é capaz de amenizar este efeito.

Como visto no Capítulo 3, o elitismo assegura que pelo menos o melhor indivíduo de uma geração ocorra na geração seguinte. Subentende-se que seja possível determinar com segurança qual será o melhor indivíduo, porém, desde que os valores objetivos sejam calculados com um número reduzido de amostras aleatoriamente posicionadas no espectro, essa premissa não pode ser mais garantida. Conseqüentemente, mesmo com o elitismo, deverá ser possível observar gerações cujo melhor indivíduo seja efetivamente inferior ao melhor indivíduo da geração anterior. Em outras palavras, certamente ocorrerá degenerações.

Para evitar degenerações e retificar o elitismo, o GA pode ser adaptado da seguinte forma. A cada geração, o suposto melhor indivíduo deverá ser reavaliado, porém empregando uma quantidade superior de amostras, de modo que a variância do valor objetivo seja

desprezível. Se esta for a primeira geração, este indivíduo reavaliado será considerado o indivíduo elitista. Caso esta geração não seja a primeira, o indivíduo reavaliado será comparado ao elitista e, caso seja melhor, deverá substituí-lo. Mas, caso o indivíduo reavaliado seja inferior ao elitista, o elitista será reinsertado na população de forma que não seja possível ocorrer degenerações. Este mecanismo pode ser estendido ao elitismo múltiplo, onde vários “melhores indivíduos” são garantidos de uma geração para a seguinte.

```

01. Crie uma população A aleatória;
02. Para cada elemento  $k$  de A:
03.      $\mathbf{A}[k].aptidão = FO(\mathbf{A}[k], S_R)$ ;
04. Ordene decrescente A;
05. Crie uma população auxiliar E de tamanho  $N_E$ ;
06. Para cada elemento  $k$  de E:
07.      $\mathbf{E}[k] = \mathbf{A}[k]$ ;
08.      $\mathbf{E}.aptidão = FO(\mathbf{A}[k], S_S)$ ;
09. Para cada geração:
10.      $k = 1$ ;
11.     Enquanto  $k \leq \mathbf{B}.tamanho$ :
12.          $\mathbf{aux} = \mathbf{A}[\text{mod}(k - 1, \mathbf{A}.tamanho) + 1]$ ;
13.          $\mathbf{B}[k] = \text{Muta\c{c}\~{a}o}(\text{Cruzamento}(\mathbf{aux}, \mathbf{A}))$ ;
14.          $\mathbf{B}[k].aptidão = FO(\mathbf{B}[k], S_R)$ ;
15.         Se  $\mathbf{B}[k]$  for melhor do que  $\mathbf{aux}$  ou  $\text{rand}() < P_B$  :
16.              $k = k + 1$ ;
17.         Para cada elemento  $k$  de A:
18.             Se  $\mathbf{A}[k].idade < idade_{max}$  ou  $idade_{max} < 0$ :
19.                 Se  $idade_{max} > 0$ :  $\mathbf{A}[k].idade = \mathbf{A}[k].idade + 1$ ;
21.                  $\mathbf{B} = \mathbf{B} + \mathbf{A}[k]$ ;
22.          $\mathbf{A} = \text{Sele\c{c}\~{a}o}(\mathbf{B})$ ;
23.         Ordene decrescente A;
24.          $k = N_E$ ;
25.         Para  $j$  iniciando em  $N_E$  e enquanto  $j \geq 1$  e  $k \geq 1$ :
26.              $\mathbf{A}[j].aptidão = FO(\mathbf{A}[j], S_S)$ ;
27.             Enquanto  $\mathbf{A}[j]$  for pior do que  $\mathbf{E}[k]$  e  $k \geq 1$ :
28.                  $k = k - 1$ ;
29.                 Se  $k \geq 1$ :  $\mathbf{E}[k] = \mathbf{A}[j]$ ;
30.         Se elitista:  $\mathbf{A} = \mathbf{A} + \mathbf{E}$ ;

```

Figura 4.12: Algoritmo para o GA flexível acelerado.

O algoritmo da Fig. 4.12 é basicamente o mesmo “algoritmo flexível” proposto no Capítulo 3. Porém, a função objetivo, representada pela função “FO”, passa a receber um

parâmetro adicional: a quantidade de amostras. O algoritmo utiliza duas quantidades diferentes, S_S e S_R . A quantidade S_S , $S_S > S_R$, é empregada apenas para reavaliar os N_E melhores indivíduos de \mathbf{A} . S_R é empregado para a avaliação ordinária. Se S_S e S_R forem sempre constantes ao longo de todo o processo de otimização, o algoritmo deverá apresentar um ganho computacional (*speed up*) de aproximadamente:

$$G_S \approx \frac{S_C \times N_B}{S_S \times N_E + S_R \times N_B} \quad (4.26)$$

onde G_S representa o ganho computacional obtido com a redução da amostragem; S_C representa a quantidade de amostras fixas e homogeneamente espaçadas empregada no GA convencional; N_B representa a capacidade da população intermediária \mathbf{B} ; S_S representa a quantidade de amostras empregadas para a avaliação dos indivíduos elitistas; S_R representa a quantidade reduzida de amostras.

Por exemplo, é razoável esperar que $S_S = S_C$. Logo, para $S_S = 200$, $S_R = 10$, $N_B = 50$ e $N_E = 1$ (valores não muito distantes da realidade), o ganho computacional G_S seria de aproximadamente 14.3, considerando obviamente $P_B = 1$ e também que, para a mesma qualidade da solução final, a mesma quantidade de gerações seja utilizada tanto para o GA convencional quanto acelerado. Na prática espera-se que a convergência do GA acelerado ainda deva ser afetada, com ganho efetivo abaixo do que o previsto através de (4.26).

Não há impedimentos para, por exemplo, aplicar a amostragem progressiva ou adaptativa para a avaliação do indivíduos elitistas. Estas técnicas podem poupar ainda mais amostras e, conseqüentemente, melhorar ainda mais o desempenho computacional do GA. Porém, nesse caso, o número de amostras S_C ou S_S seriam realmente desconhecidos, impossibilitando o uso de (4.26) para a previsão de algum ganho computacional.

4.3.4.2 Amostragem aleatória aplicada ao EPSO: o EPSO acelerado

A amostragem aleatória reduzida também pode melhorar o desempenho computacional do EPSO. Similarmente ao GA, as chamadas à FO passam a considerar o número de amostras como parâmetro adicional. Também reavaliações de partículas precisam ser efetuadas em certos pontos estratégicos do algoritmo. A Fig. 4.13 exibe o algoritmo modificado para o EPSO flexível introduzido no Capítulo 3.

```

01. Cria um enxame de partículas A aleatórias;
02. Para cada elemento  $k$  de A:
03.      $\mathbf{A}[k].\text{aptidão} = \text{FO}(\mathbf{A}[k]);$ 
04. B = A;
05. A.melhor.aptidão =  $\text{FO}(\mathbf{A.melhor}, S_S);$ 
06. global = A.melhor;
07. Para cada iteração:
08.     Para cada partícula  $k$  de A:
09.         aux1 = global;
10.         Se topologia for local e  $k > 1$ :
11.             Decremente  $j$  de  $k - 1$  até 1 e enquanto B[ $j$ ] não for melhor que B[ $k$ ];
12.             Se B[ $j$ ] for melhor que B[ $k$ ]: aux1 = B[ $j$ ];
13.         aux2 = A[ $k$ ];
14.         A[ $k$ ] =  $\text{Movimento}(\mathbf{A}[k], \mathbf{aux1}, \mathbf{B}[k]);$ 
15.          $\mathbf{A}[k].\text{aptidão} = \text{FO}(\mathbf{A}[k], S_{Ri});$ 
16.         Repita  $N_\lambda$  vezes:
17.             aux3 =  $\text{Movimento}(\text{Muta\c{c}o\~{a}o}(\mathbf{aux2}), \mathbf{aux1}, \mathbf{B}[k]);$ 
18.              $\mathbf{aux3}.\text{aptidão} = \text{FO}(\mathbf{aux3}, S_{Ri});$ 
19.             Se aux3 for melhor do que A[ $k$ ]:
20.                 A[ $k$ ] = aux3;
21.              $\mathbf{A}[k].\text{aptidão} = \text{FO}(\mathbf{A}[k], S_{Re});$ 
22.             Se A[ $k$ ] for melhor do que B[ $k$ ]: B[ $k$ ] = A[ $k$ ];
23.             Se  $k > 1$ :
24.                 Se B[ $k$ ] for melhor do que B[ $k - 1$ ]:
25.                     Permute A[ $k$ ] e A[ $k - 1$ ], permute B[ $k$ ] e B[ $k - 1$ ];
26.             B.melhor.aptidão =  $\text{FO}(\mathbf{B.melhor}, S_S);$ 
27.             Se B.melhor for superior global: global = B.melhor;
28.             Se a  $\text{Diversidade}(\mathbf{A}) < \text{mínimo}$ :
29.                 Para  $k$  iniciando em  $\mathbf{A.tamanho}/2$  até  $\mathbf{A.tamanho}$ :
30.                     B[ $k$ ] = A[ $k$ ] = “Nova partícula aleatória”;
31.                      $\mathbf{B}[k].\text{aptidão} = \mathbf{A}[k].\text{aptidão} = \text{FO}(\mathbf{A}[k]);$ 

```

Figura 4.13: Algoritmo para o EPSO flexível

O algoritmo da Fig. 4.13 na verdade exhibe três diferentes amostragens, empregando S_S , S_R e S_{Ri} . A quantidade S_S é empregada apenas para a reavaliação da suposta melhor partícula do enxame, sendo idealmente capaz de garantir erros desprezíveis para o cálculo do valor objetivo. S_R é empregado na avaliação das partículas ordinárias após seu movimento. S_{Ri} está associado exclusivamente ao EPSO, sendo empregado na avaliação das partículas temporárias geradas em prol do processo auto-adaptativo. Para $S_{Ri} = S_R$ a linha 21 pode ser eliminada pois a reavaliação de $\mathbf{A}[k]$ seria redundante. Para $N_\lambda = 0$, o mecanismo auto-adaptativo do EPSO seria desligado, tornando a linha 15 também redundante. A expressão para o ganho de amostragem esperado G_S precisa considerar todos estes cenários:

$$G_S = \begin{cases} \frac{S_S N_B}{S_S + S_R \times N_B} & se & N_\lambda = 0 \\ \frac{S_S (N_\lambda + 1) N_B}{S_S + S_{Ri} (N_\lambda + 1) N_B} & se & S_{Ri} = S_R \\ \frac{S_S (N_\lambda + 1) N_B}{S_S + [S_R + S_{Ri} (N_\lambda + 1)] N_B} & se & N_\lambda \neq 0 \text{ ou } S_{Ri} \neq S_R \end{cases} \quad (4.27)$$

Capítulo 5

Resultados

No Capítulo anterior foram mostradas as contribuições do autor. Tratavam-se de duas classes de técnicas: as relacionadas à decodificação e as relacionadas à aceleração computacional (através de estratégias especiais de amostragem). Este Capítulo tem como objetivo demonstrar e validar todas estas contribuições.

Para tanto, a primeira seção deste Capítulo procura investigar a combinação ideal dos parâmetros básicos tanto para o GA quanto para o PSO. As seções subsequentes procuram investigar o impacto das técnicas de decodificação e das técnicas de aceleração em ambas as heurísticas. A última seção procura discutir as diferenças, vantagens e desvantagens da aplicação das técnicas desenvolvidas para o PSO e para o GA.

5.1 GA: estudo das técnicas de decodificação e de aceleração

5.1.1 Parâmetros básicos do GA

É importante determinar os parâmetros básicos do GA aplicá-lo ao problema de síntese de grades em fibra. Por exemplo, o número mais adequado de indivíduos, o tipo de seleção e a pressão seletiva, o tipo de cruzamento e seus eventuais parâmetros e o tipo de mutação e seus eventuais parâmetros. Além disso, existem variações do elitismo que podem ser aplicadas ao GA e estratégias adaptativas ou auto-adaptativas. O problema certamente poderá ser solucionado de forma mais eficiente com o conjunto apropriado de parâmetros ou estratégias, porém o conjunto deles é vasto e o próprio problema de síntese de grades pode demandar uma

combinação diferente destes parâmetros gerais para cada projeto de FBG diferente. Portanto, a determinação rigorosa de todos os parâmetros para o GA.

Porém, a literatura celebra certas combinações fundamentais de parâmetros, métodos e técnicas ideais, consagradas pelas inúmeras aplicações e sucessos em problemas diversificados ou comparáveis. Por exemplo, para o GA, a seleção por torneio é considerada pelo menos tão eficiente quanto à proporcional, sendo de fato uma das melhores técnicas disponíveis. Outro exemplo é a comparação entre as técnicas de cruzamento BLX, SBX, ou o cruzamento empregando lógica nebulosa (lógica Fuzzy)[35], que em geral resultam equivalentes: quaisquer ganhos obtidos com a escolha precisa de uma ou de outra destas técnicas de cruzamento provavelmente levaria a ganhos poucos expressivos. Portanto, prefere-se BLX pela simplicidade. Finalmente, é fato que técnicas auto-adaptativas para controle de parâmetros estratégicos são superiores às outras técnicas dinâmicas ou adaptativas [25].

Portanto, não é preciso determinar rigorosamente todos os parâmetros e técnicas para o GA e para o PSO. É possível concentrar esforços na determinação de parâmetros realmente importantes: o tamanho da população, a pressão seletiva e os parâmetros associados às técnicas adaptativas ou auto-adaptativas para atualização dos parâmetros do cruzamento, mutação e os pesos do PSO.

5.1.1.1 GA: tamanho da população

Dentre todos os parâmetros, talvez o tamanho da população para o GA possa ser considerado o parâmetro mais crítico, cujo dimensionamento costuma ser empírico. Embora possa ser superdimensionado com alguma segurança, um valor menor para o tamanho da população freqüentemente é preferível em prol da redução do custo computacional do processo de otimização.

De modo geral, o tamanho da população influencia de forma bastante contundente o desempenho geral do GA. Pode-se afirmar que, para uma heurística empregando valores clássicos da literatura para os todos os parâmetros restantes, um tamanho de população cada vez mais próximo do ideal resultará em um desempenho cada vez superior ou, em outras palavras, pode-se afirmar que seja possível determinar o tamanho da população de forma independente.

Antes de prosseguir, entretanto, é necessário conceber um projeto de FBG adequado às experimentações associadas ao dimensionamento dos parâmetros básicos. Para tal, considere um filtro refletor simples para a faixa de comprimentos de onda de 1549.8nm a 1550.2nm (refletividade nula fora dessa faixa). Para este projeto, deseja-se determinar a visibilidade

de franjas para uma FBG de 2.5cm de comprimento dividida em 50 seções uniformes. A codificação real convencional considera apenas os valores de ν_i para $i = 1, 2, 3, \dots, 50$ com valor mínimo $\nu_i^{min} = 0$ e máximo $\nu_i^{max} = 4 \times 10^8$. Os demais parâmetros são constantes: $\delta n_{effi} = 10^{-12}$, $\Lambda_i = 1550nm/(2n_{eff})$, $\Delta z = 200\mu m$ e $\phi_i = 0$.

Para o projeto deste filtro refletor, foram realizadas simulações empregando diversos tamanhos de população: 10, 20 e 40 indivíduos. Utilizou-se tanto a seleção elitista como a seleção por torneio estocástico. Para a seleção elitista, sempre 50% melhores indivíduos da população foram conservados para a geração seguinte. Para a seleção por torneio foram utilizados tamanhos (ou capacidades) 3 e 5 para populações de 10 indivíduos, 5 e 10 para populações de 20 indivíduos, e tamanhos de 5, 10 e 20 para populações de 40. Dois tipos de elitismo foram utilizados: juntamente com a seleção elitista, o elitismo tipicamente empregado nas estratégias evolucionárias, onde os indivíduos progenitores sobrevivem e disputam juntamente com os seus filhos pela seleção; o elitismo clássico tipicamente adotado para o GA, onde o melhor indivíduo da população atual tem presença garantida na população da geração subsequente.

Os demais parâmetros do GA foram mantidos constantes. A mutação gaussiana com probabilidade de mutação de 100% e desvio padrão dinâmico dado pela expressão $\sigma_M = 0.1 - 0.099[(g - 1)/(G - 1)]^{0.2}$, onde g e G representam respectivamente o número da geração atual e o número máximo de gerações. Cruzamento adotado foi o BLX com probabilidade P_C igual a 50% e parâmetro α igual a 0.5.

A Tabela 5.1 exhibe, respectivamente nas colunas 4, 5 e 6, os valores de desvio padrão, valores médios e mínimos (mínimo = “melhor”) calculados com base nos valores FO finais após 10000 gerações, sendo a FO baseada no cálculo do erro RMS. Para cada combinação de parâmetros o GA foi executado 10 vezes. Na primeira coluna, utilizou-se três códigos para indicar o tipo de elitismo empregado: “A”, “B” e “C”. “A” para o elitismo clássico onde o melhor indivíduo é garantido na geração seguinte, “B” indica o uso da seleção elitista onde os 50% melhores indivíduos são selecionados para a geração seguinte e “C” indica o uso da seleção por torneio sem elitismo algum. A segunda coluna da tabela mostra o tamanho da população. A terceira indica o tipo de seleção, “elitista” ou “torneio”, sendo que para este último a capacidade do torneio é exibida após “/”. A última coluna exhibe a convergência média em número de gerações, uma métrica definida como o número da geração G_c acima da qual o erro entre o valor da FO (em G_c) e valor FO final (em $G = 10000$) é menor que 1%, isto é:

$$\frac{|FO_c - FO_G|}{FO_G} = 0.01 \quad (5.1)$$

onde FO_c é o valor da função objetivo da geração G_c e FO_G é o valor da função objetivo da

última geração G .

O número máximo de gerações G igual a 10000 foi escolhido por ser suficientemente grande para garantir que o GA possa alcançar um valor mínimo estável para a FO. A Fig. 5.1 mostra um exemplo onde $G_c = 6825$, $FO_c = 0.8677$ e $FO_G = 0.8591$: a diferença entre os valores da FO nestes dois pontos é suficientemente pequena para satisfazer 5.1.

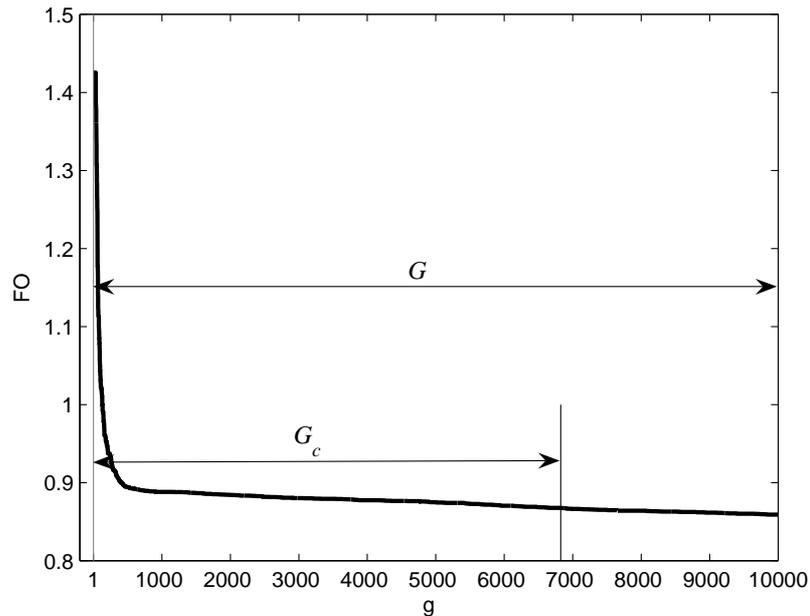


Figura 5.1: Exemplo de uma curva para o valor de FO em função do número de gerações, sendo o “número de geração de convergência” (G_c) posto em destaque.

Os números mostrados na Tabela 5.1 claramente indicam melhores resultados finais para populações de 20 e 40 indivíduos. Os valores médios da FO entretanto, aproximam-se mais dos valores mínimos (coluna “Melhor”) para populações com 40 indivíduos. Os resultados também mostram que a seleção por torneio com pressões seletivas elevadas, isto é, tamanhos de 25% e 50% da população, parecem oferecer melhores resultados. Porém nada se pode afirmar sobre o efeito do tipo de elitismo.

5.1.1.2 Elitismo, Seleção e parâmetros correlatos

Em busca de resultados mais conclusivos a cerca do elitismo e técnicas de seleção, o GA, usando mutação com desvio auto-adaptativo isotrópico com $\tau_M = 0.1$, foi utilizado para preencher uma nova tabela de valores. A Tabela 5.2 exhibe valores que mais conclusivamente apontam um tamanho de população igual a 40 como bastante apropriado, visto que os valores

Elitismo	População	Seleção/ Capacidade	FO			Convergência
			Desvio	Média	Melhor	média
C	10	Torneio/3	0.064898	1.07739	0.99194	942.8
A	10	Torneio/3	0.120880	0.98890	0.87325	3264.3
A	10	Torneio/5	0.061417	0.95009	0.86862	1144.9
C	10	Torneio/5	0.061648	0.93122	0.85959	1661.4
B	10	Elitista	0.072513	0.77274	0.56902	5854.9
A	20	Torneio/5	0.037011	0.91095	0.87245	1398.4
C	20	Torneio/5	0.246028	0.80294	0.33240	2107.0
B	20	Elitista	0.145126	0.73535	0.32384	6998.3
A	20	Torneio/10	0.269748	0.72017	0.31870	2589.6
C	20	Torneio/10	0.261432	0.71780	0.32803	2319.6
A	40	Torneio/5	0.243030	0.71135	0.33715	2517.5
C	40	Torneio/5	0.274450	0.66259	0.32595	2925.6
B	40	Elitista	0.218420	0.58372	0.32478	8493.8
A	40	Torneio/10	0.220310	0.45066	0.33206	2896.0
C	40	Torneio/20	0.214060	0.44311	0.33339	2814.8
C	40	Torneio/10	0.159220	0.40019	0.31803	3447.5
A	40	Torneio/20	0.167770	0.39827	0.33197	2708.6

Tabela 5.1: Tabela de experimentações para a determinação dos principais parâmetros do GA.

FO médios se aproximam muito mais dos respectivos valores mínimos para populações com este tamanho do que para populações com 10 e 20 indivíduos. Mas outras importantes observações podem ser obtidas através desta tabela: a seleção elitista parece ser mais eficiente com razões filhos/pais elevadas, algo próximo de 5, e, principalmente, sem a sobrevivência dos pais (sem elitismo). Aliás, de um modo geral a ausência de qualquer forma de elitismo parece oferecer melhores resultados seja o tipo de seleção elitista ou por torneio.

De um modo geral a Tabela 5.2 confirma as observações da literatura [25], como a superioridade de esquemas auto-adaptativos frente a esquemas dinâmicos (como o utilizado na Tabela 5.1). Os resultados também indicam que a seleção elitista com altas razões filhos/pais parece oferecer bons resultados quando combinada com estratégias auto-adaptativas. De fato, este tipo de seleção parece ser sutilmente superior em relação à seleção por torneio também quando associado à técnicas adaptativas, como mostrado na Tabela 5.3. Esta tabela compara os resultados para o GA não elitista para diferentes estratégias de adaptação do parâmetro

Elitismo	População		FO			Convergência
	Pais/Filhos	Seleção/ Capacidade	Desvio	Média	Melhor	média
B	2/10	Elitista	0.006292	0.79705	0.78772	2437.3
A	10/10	Torneio/5	0.010817	0.78434	0.76187	2328.2
B	10/10	Elitista	0.043901	0.77701	0.65314	4161.2
C	2/10	Elitista	0.041144	0.76917	0.67176	2891.9
B	4/20	Elitista	0.010528	0.78625	0.76671	2052.4
B	20/20	Elitista	0.006897	0.78448	0.77239	3809.3
C	20/20	Torneio/5	0.176057	0.63927	0.32628	6397.8
A	20/20	Torneio/5	0.149617	0.53222	0.32237	9827.7
C	20/20	Torneio/10	0.168930	0.49765	0.31889	8289.6
A	20/20	Torneio/10	0.175900	0.48874	0.32066	9745.8
C	4/20	Elitista	0.140523	0.38717	0.32088	9311.9
B	8/40	Elitista	0.011903	0.78531	0.75540	2886.2
A	40/40	Torneio/10	0.003631	0.32037	0.31819	7780.7
A	40/40	Torneio/20	0.000924	0.31953	0.31829	8076.2
C	40/40	Torneio/20	0.000725	0.31908	0.31817	7839.9
C	40/40	Torneio/10	0.000613	0.31890	0.31810	7165.7
C	8/40	Elitista	0.000732	0.31885	0.31791	6870.7

Tabela 5.2: Tabela de experimentações para a determinação dos principais parâmetros do GA.

σ_M . Tal como esperado, as técnicas dinâmicas superaram as estáticas, que por sua vez foram superadas pelas adaptativas e auto-adaptativas. Particularmente destaca-se o desempenho do *adaptive* GA (AGA), que obteve resultados compatíveis aos das estratégias auto-adaptativas, porém com uma convergência aparentemente mais rápida. O AGA com *ranking* utilizou $\rho_{C1} = \rho_{C2} = 1$ (valores clássicos na literatura) para o controle da probabilidade de cruzamento P_C e $\rho_{M1} = \rho_{M2} = 0.01$ (valor determinado empiricamente) para o controle de σ_M . A estratégia auto-adaptativa isotrópica empregada utilizou $\tau_M = 0.1$.

5.1.1.3 Mutaç o com desvio auto-adaptativo e o par metro τ_M

Para estrat gias evolucion rias que utilizam apenas a muta o sobre um indiv duo para gerar descendentes, a literatura estabelece que $\tau_M \propto 1/\sqrt{D}$, onde D representa o n mero de dimens es do problema. Nas tabelas 5.2 e 5.3 foi utilizado $\tau_M = 0.1$, algo de fato pr ximo de

Adaptação de σ_M	População Pais/Filhos	Seleção/ Capacidade	FO			Convergência
			Desvio	Média	Melhor	média
Valor fixo 0.001	8/40	Elitista	0.009226	0.79741	0.78172	4487.6
Valor fixo 0.001	40/40	Torneio/10	0.008141	0.78441	0.77499	5779.8
Dinâmico	8/40	Elitista	0.267150	0.55579	0.33676	2858.7
Dinâmico	40/40	Torneio/10	0.159220	0.40019	0.31803	3447.5
Regra de 1/5	40/40	Torneio/10	0.068106	0.35287	0.31900	9156.4
Regra de 1/5	8/40	Elitista	0.000699	0.32538	0.32392	7645.7
AGA	40/40	Torneio/10	0.000929	0.32354	0.32255	1971.3
AGA	8/40	Elitista	0.000375	0.31911	0.31847	2020.3
Auto-adaptativo	40/40	Torneio/10	0.000613	0.31890	0.31810	7165.7
Auto-adaptativo	8/40	Elitista	0.000732	0.31885	0.31791	6870.7

Tabela 5.3: Tabela de experimentações para a comparação entre as diversas técnicas de adaptação da mutação.

$1/\sqrt{50}$ ($D = 50$). Porém, o GA utilizado fez uso de cruzamento com 8 indivíduos para criar 40 descendentes a cada geração. A relação de proporcionalidade entre τ_M e \sqrt{D} é algo precário para estratégias evolucionárias sem cruzamento, ainda mais o é para o GA adotado. De qualquer modo não há como determinar o valor mais apropriado de τ_M senão empiricamente. Para este propósito o GA foi executado 10 vezes para diversos valores deste parâmetro. A Fig. 5.2 exhibe os valores médios, mínimos e de desvio padrão para a FO em função de τ_M . O melhor desempenho para o GA está associado ao valor τ_M de aproximadamente 0.025.

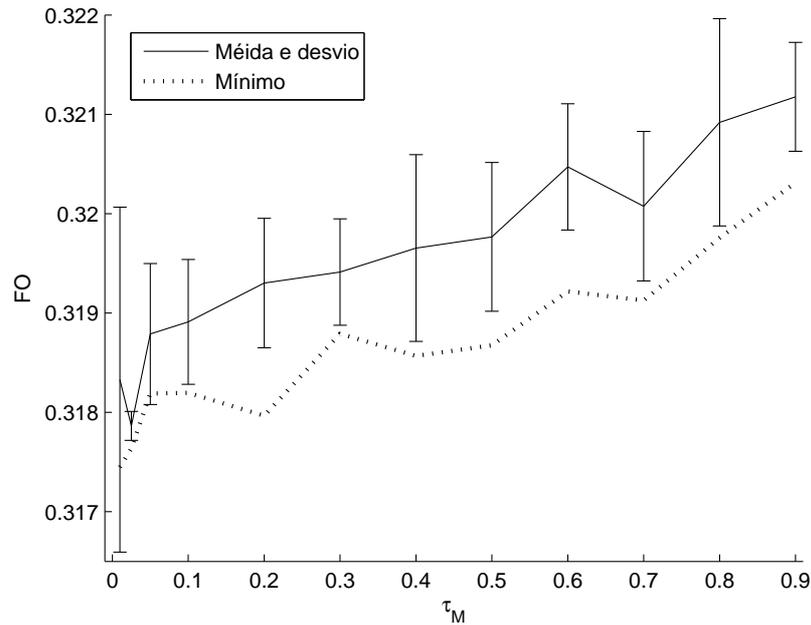


Figura 5.2: Variação do valor FO médio, desvio padrão (mostrado em barras) e valor mínimo em função de τ_M .

Se fosse possível estabelecer uma relação de simples proporcionalidade entre τ_M e $1/\sqrt{D}$, especificamente para este problema de síntese de filtros ópticos e considerando $\tau_M = 0.025$ como ideal para $D = 50$, a expressão capaz de associar τ_M a $1/\sqrt{D}$ seria:

$$\tau_M \approx \frac{0.17678}{\sqrt{D}}. \quad (5.2)$$

5.1.1.4 Mutação com desvio auto-adaptativo e cruzamento adaptativo com AGA

Os melhores resultados para o GA para o presente problema foram conseguidos utilizando auto-adaptação de σ_M . Entretanto, o AGA, uma técnica adaptativa notavelmente simples, consegue resultados bastante interessantes. É importante observar que a técnica adaptativa AGA atua não apenas em algum parâmetro de controle da mutação, mas também sobre o valor de P_C . Este fato permite a conceber uma versão híbrida do GA, onde o AGA providencia a adaptação de P_C apenas, enquanto σ_M é adaptado através da técnica auto-adaptativa isotrópica. Para verificar a capacidade de um possível GA adaptativo-auto-adaptativo, a Tabela 5.4 exibe os valores de FO (desvio, médio e mínimo) e número de gerações médio para a convergência para o GA AGA, auto-adaptativo e híbrido, todos estes esquemas tendo como base o GA com seleção elitista, com 8 pais e 40 filhos por geração.

Adaptação de σ_M	FO			Convergência
	Desvio	Média	Melhor	média
AGA	0.000375	0.31911	0.31847	2020.3
Auto $\tau_M = 0.05$	0.000710	0.31879	0.31819	7153.6
Híbrido $\tau_M = 0.025$	0.000378	0.31810	0.31754	4443.6
Híbrido $\tau_M = 0.05$	0.000318	0.31797	0.31763	3996.7
Auto $\tau_M = 0.025$	0.000146	0.31786	0.31764	4352.0

Tabela 5.4: Comparação entre o esquema AGA puro, auto-adaptativo puro e híbrido.

A Tabela 5.4 revela uma grande similaridade entre o comportamento dos três algoritmos, o AGA puro (empregando os mesmos parâmetros utilizados na Tabela 5.3) o GA híbrido AGA-auto-adaptativo e o GA apenas auto-adaptativo. Porém, o GA híbrido e o auto-adaptativo puro foram aplicados utilizando dois valores diferentes para τ_M : 0.025 e 0.05. O desvio padrão médio e a convergência média revelam um impacto expressivo da variação do parâmetro τ_M sobre o GA auto-adaptativo puro, algo muito mais acentuado do que o observado para o GA híbrido. De fato, para diferentes valores de τ_M a versão híbrida oferece resultados praticamente indistinguíveis: é possível que a aplicação do AGA para a adaptação de P_C ofereça mais robustez à heurística.

5.1.1.5 Alvo com tolerâncias

A Fig. 5.3 exibe a curva de refletividade para o melhor resultado da Tabela 5.4, conseguido para o GA com seleção elitista de 8 pais e 40 filhos, empregando mutação com desvio auto-adaptativo com $\tau_M = 0.1$. A Fig. 5.3 também exibe o perfil de apodização para o parâmetro ν em função do número da seção uniforme obtido pelo GA. É interessante observar o aspecto da apodização, que resulta similar ao de uma superestrutura, isto é, uma seqüência de grades intercaladas por espaços em branco. Porém os valores de ν nestes espaços não são desprezíveis, variando entre um máximo valor no centro na FBG e um valor mínimo nas bordas, acompanhando um padrão que lembra bastante o comportamento de uma gaussiana. Esta curva de apodização curiosa encontrada pelo GA é provavelmente fruto das rigorosas restrições impostas pelo alvo em formato pulso centrado em 1550nm e com largura de 0.4nm. Este alvo estabelece transições súbitas nas bordas da banda de máxima refletividade, entre valores de 0 e 100%. O GA, na medida do possível, procura por uma apodização capaz de tanto oferecer transições bruscas entre refletividades mínima e máxima tanto reduzir ao máximo os lóbulos laterais. Obviamente não é possível atender ambas as restrições de forma

razoável. Portanto, um alvo mais realista precisa considerar uma região de tolerância em torno de 1549.8nm e 1550.2nm, onde, por exemplo, a função objetivo não deverá considerar quaisquer amostras.

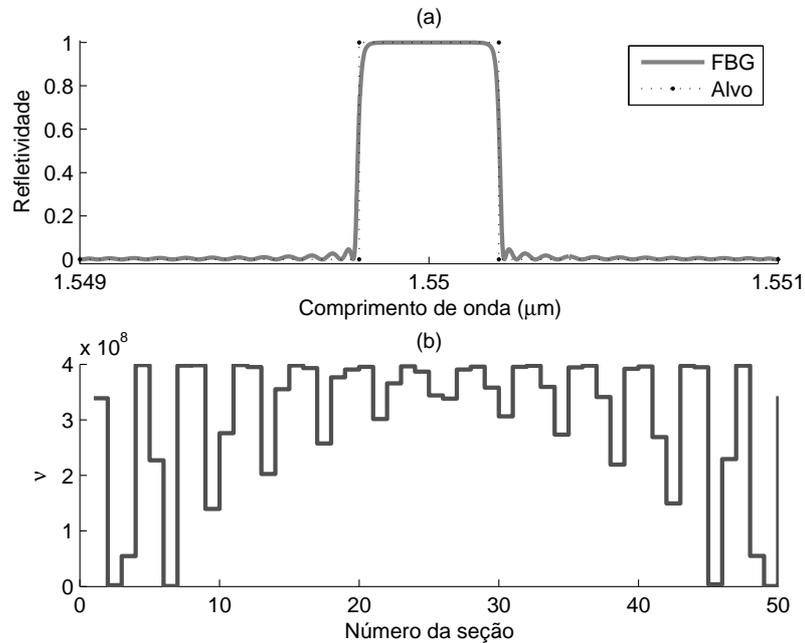


Figura 5.3: Curva de refletividade para FBG sintetizada e alvo (a); curva de apodização obtida para a visibilidade de franjas ν (b).

Considere, por exemplo, regiões de tolerância de 0.1nm centradas em 1549.8nm e 1550.2nm, suficientes para permitir a transição da refletividade de 0 a 100%. Portanto, o alvo poderia ser representado por três intervalos: $[1549\text{nm}; 1549.75\text{nm}]$ e $[1550.25\text{nm}; 1551\text{nm}]$ onde refletividade seria nula; e $[1549.85\text{nm}; 1550.15\text{nm}]$ onde a refletividade seria máxima. A Fig. 5.4 exibe o alvo com tolerâncias, cujas dimensões dos intervalos são de fato bastante consideráveis em relação a faixa de máxima refletividade.

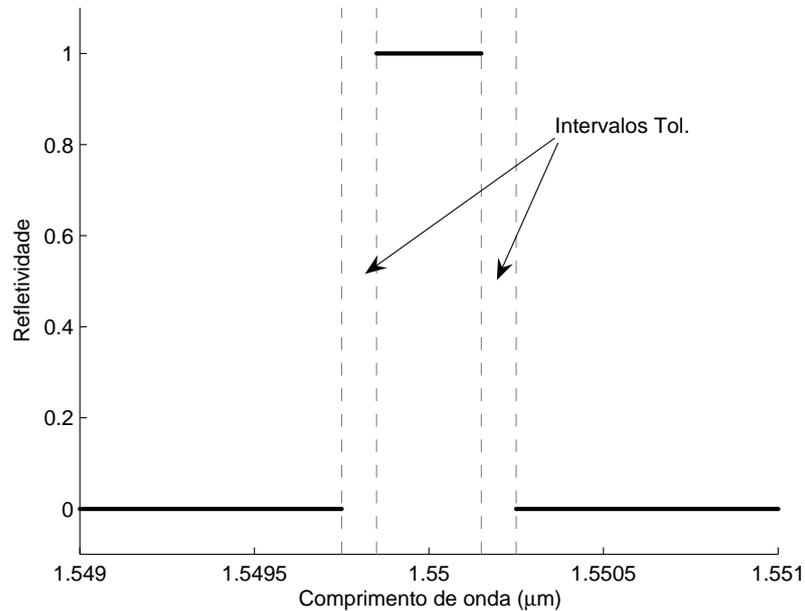


Figura 5.4: Curva alvo respeitando intervalos de tolerância demarcado através de linhas tracejadas horizontais.

A Fig. 5.5 compara o espectro de refletividade e a curva de apodização de uma FBG típica obtida pelo GA utilizando o alvo melhorado com tolerâncias, mostrado na Fig. 5.4, com a refletividade e apodização mostrados na Fig. 5.3 obtidos para uma FBG típica encontrada pelo GA empregando o alvo sem tolerâncias. Com o alvo melhorado, a curva de refletividade (em dB) resulta com lóbulos laterais muito inferiores aos observados pela curva de refletividade obtida utilizando-se o alvo original. Por outro lado, como esperado, observa-se também transições mais lentas nas bordas da banda de máxima refletividade. É interessante observar também que o perfil de apodização para ν , obtido usando o alvo com tolerâncias, é mais suave e desprovido de trechos máximos periódicos típicos de superestruturas.

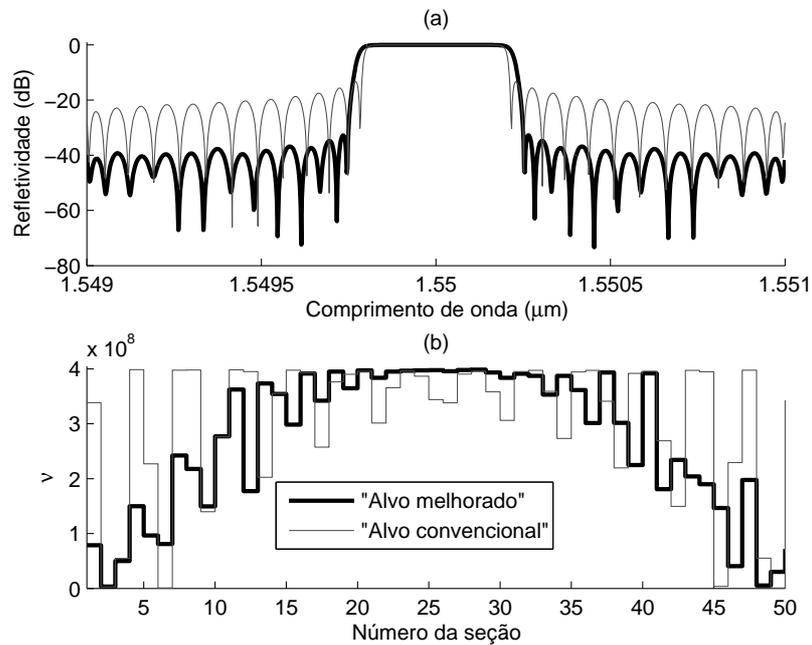


Figura 5.5: Curvas de refletividade (a) e de apodização da visibilidade de franjas ν (b) para uma FBG sintetizada utilizando o alvo original e para outra obtida com o alvo com tolerâncias.

Nas próximas subseções este alvo melhorado com tolerâncias será empregado no lugar do alvo original tanto para os exemplos envolvendo o GA quanto os envolvendo o PSO, pois permitirá demonstrar e explorar eventuais vantagens das técnicas de decodificação suavizadas. Utilizando o alvo original o GA seria possível obter os melhores resultados apenas para a decodificação convencional, uma vez que as outras técnicas de decodificação propostas sempre procuram reduzir os graus de liberdade dos perfis, o que impediria por completo apodizações similares a superestruturas (supostamente mais vantajosas para obter transições abruptas no espectro de refletividade).

5.1.2 A decodificação suavizada

As técnicas de decodificação suavizadas propostas no capítulo anterior operam reduzindo ou limitando as discrepâncias entre os parâmetros de duas seções uniformes contíguas. Uma vez que não alteram a dimensionalidade do problema, considera-se que todos os parâmetros supostamente ideais determinados empiricamente nas subseções anteriores deste capítulo deverão permanecer similares.

Para o mesmo projeto de filtro refletor simples empregado nos exemplos anteriores e para o GA utilizando mutação com desvio auto-adaptativo com $\tau_m = 0.025$, 8 progenitores

e 40 descendentes por geração, a Tabela 5.5 exhibe resultados obtidos utilizando decodificação convencional, DSR com $p = \{0.02, 0.05, 0.1\}$ e utilizando a DSMM com $span = \{2, 5, 10\}$.

Decodificação	FO			Convergência
	Desvio	Média	Melhor	média
$p = 0.02$	0.0007689	0.2854342	0.2847658	2065.6
$p = 0.05$	0.0002808	0.0225848	0.0220678	2569.3
$span = 10$	0.0000605	0.0225507	0.0224871	1758.2
$span = 2$	0.0000636	0.0037270	0.0036410	2298.0
$span = 5$	0.0000588	0.0034668	0.0033949	4445.4
$p = 0.1$	0.0000404	0.0033735	0.0033189	4392.0
Não suavizado	0.0000606	0.0031215	0.0030611	3220.8

Tabela 5.5: Comparação entre o GA utilizando DSMM para três diferentes valores de $span$, o GA utilizando DSR para três valores de p e o GA não suavizado.

É interessante observar na Tabela 5.5 o considerável sucesso do GA utilizando decodificação convencional frente ao GA utilizando decodificação suavizada. Embora a suavização seja capaz de filtrar e simplificar o espaço de buscas, permitindo, teoricamente, uma aceleração do processo de busca, o GA com decodificação convencional obtém resultados melhores a custo de menos gerações. É possível que a filtragem do espaço de buscas produza como efeito colateral o ocultamento de soluções promissoras, as quais, portanto, são visíveis apenas ao GA com decodificação convencional. Além disso, a força da suavização (diretamente proporcional a $span$ e inversamente proporcional a p) para $p = 0.1$ não parece ser suficiente para simplificar o espaço de buscas a ponto de acelerar substancialmente a otimização. De fato, simplificações mais radicais do espaço de buscas, conseguidas com suavização utilizando $span = 10$ ou $p = 0.02$, permitem convergências consideravelmente mais rápidas.

A Fig. 5.6 exhibe as curvas de refletividade e os perfis de ν obtidos utilizando DSR para $p = 0.1$ e utilizando decodificação convencional. É interessante observar que praticamente não existem diferenças entre as curvas de refletividade: ambas apresentam lóbulos laterais inferiores a -30dB. Porém as diferenças são substanciais para o perfil de apodização de ν . De fato, para a FBG sintetizada utilizando decodificação convencional, o aspecto da apodização é mais caótico e descontínuo. A apodização conseguida com a decodificação suavizada assemelha-se a uma curva média em relação a curva de apodização não suavizada. É óbvio que as soluções são bastante similares, porém a superioridade da FBG não suavizada, expressa em números na Tabela 5.5, demonstra que mesmo as perturbações “caóticas” da sua apodização possuem algum propósito. Do ponto de vista prático, entretanto, é sempre preferível apodizações mais

simples, sendo essa a principal vantagem das técnicas de decodificação suavizadas.

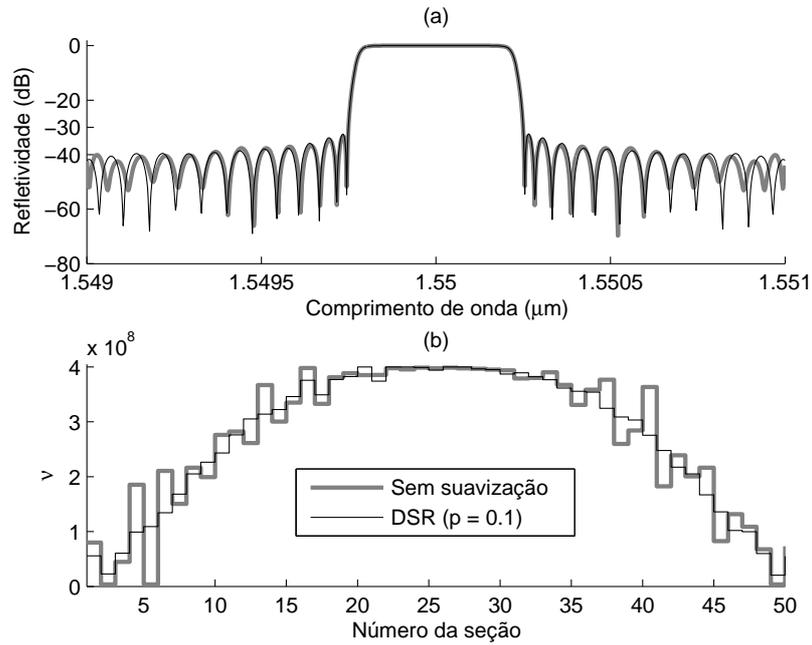


Figura 5.6: Curvas de refletividade em dB (a) e curvas e perfis de apodização (b) para ν de grades obtidas com decodificação convencional e DSR ($p = 0.1$).

A Fig. 5.7 compara três resultados típicos obtidos pelo GA utilizando DSR com os valores de $p = 0.02$, $p = 0.05$ e $p = 0.1$. Os resultados exibem alguma progressividade no que diz respeito a intensidade da suavização. A Fig. 5.8 também compara três resultados típicos obtidos com suavização, porém utilizando a DSMM para $span = 2$, 5 e 10. Diferentemente da decodificação restritiva, os valores de $span$ não parecem exercer uma influência muito linear sobre a intensidade da suavização, visto que para $span = 2$ e $span = 5$ os resultados são bastante similares. De fato, p implica que ν_i pode ser diferente de ν_{i+1} ou ν_{i-1} de no máximo $p(\nu_{max} - \nu_{min})$ (onde i é o número da seção), uma relação direta, proporcional e certamente muito mais óbvia entre o parâmetro de suavização e as apodizações esperadas. Por outro lado, não se pode estabelecer um tipo de relação como esse entre $span$ e o aspecto da curva apodização resultante. Certamente aí está uma vantagem da DSR sobre a DSMM.

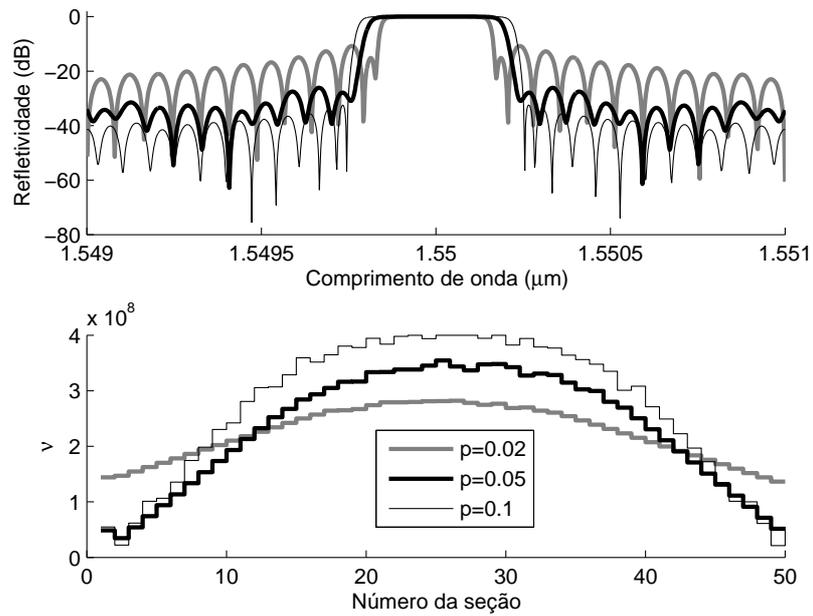


Figura 5.7: Curvas de refletividade em dB (a) e curvas e perfis de apodização (b) para ν de grades obtidas com DSR para $p = 0.02, 0.05$ e 0.1 .

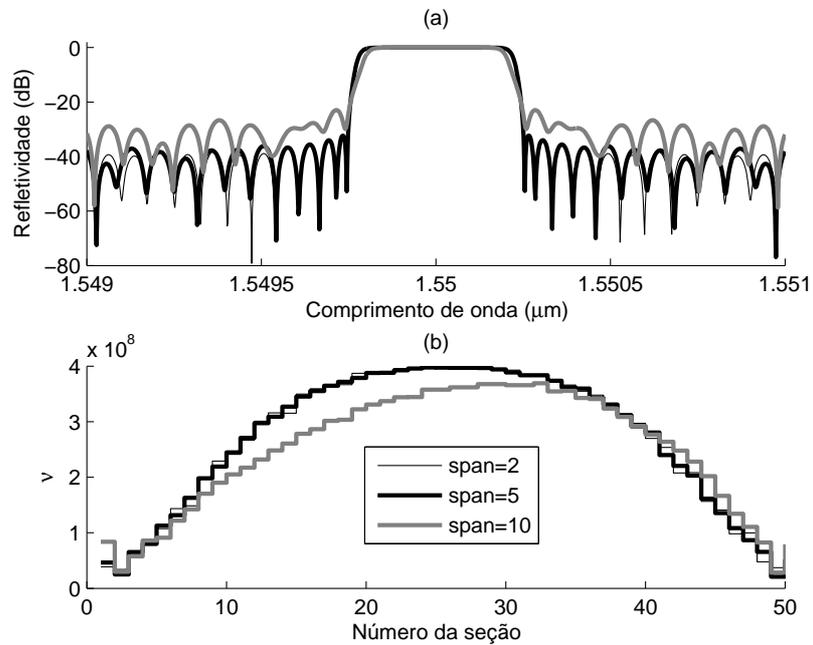


Figura 5.8: Curvas de refletividade em dB (a) e curvas e perfis de apodização (b) para ν de grades obtidas com DSMM para $span = 2, 5$ e 10 .

5.1.2.1 Decodificação suavizada para a síntese de filtros triangulares

Um filtro triangular (ou rampa) é um filtro cuja refletividade varia linearmente dentro de uma faixa do espectro. Constitui um tipo de componente especial bastante interessante para a composição de redes de sensores ópticos. As FBG apodizadas ordinárias ou uniformes são suficientes como transdutores de grandezas físicas, como a tensão mecânica e temperatura, convertendo estas grandezas em deslocamentos espectrais de picos de reflexão. Porém exigem do outro lado da rede um sistema de interrogação, responsável também pela conversão destes deslocamentos espectrais em sinais úteis. Um analisador de espectro óptico é apropriado para este propósito, porém trata-se de um equipamento optoeletrônico dispendioso e relativamente frágil. Para as aplicações em regiões inóspitas e por outras questões é interessante manter, o quanto possível, o processamento do sinal no domínio óptico e, principalmente, utilizar simples medidores de potência para a aferição. Por esta razão, é comum a utilização de filtros de borda nos aparatos interrogadores, que realizam a interseção do sinal refletido pela rede de sensores FBG com um filtro refletor cuidadosamente alinhado espectralmente de modo a refletir (ou transmitir) tanto mais potência quanto maior o deslocamento dos sinais no espectro. O filtro rampa substitui com vantagens os filtros de borda convencionais baseados em FBGs simples, pois possuem muito mais largura espectral e propiciam mais precisão nas leituras graças a uma conversão linear do deslocamento espectral para potência óptica [10].

O filtro rampa pode ser bastante útil para testar as técnicas de decodificação suavizadas pois trata-se de um problema mais difícil: exige que mais do que um parâmetro da FBG sejam simultaneamente determinados pelo GA, geralmente a apodização de ν ou δn_{eff} e a variação de λ_B ou Λ em função do comprimento da grade. Esta exigência está relacionada com, dentre outras coisas, a largura do espectro deste tipo de filtro, que costuma exceder 0.5nm.

Considere o seguinte projeto: um filtro triangular cuja a refletividade varie linearmente de zero em 1549.5nm a 1 em 1550.5nm, não interessando o aspecto da curva de refletividade fora deste intervalo. Deseja-se que o GA determine a apodização de ν e λ_B , sendo $0 \leq \nu \leq 4 \times 10^8$ e $1549\text{nm} \leq \lambda_B \leq 1551\text{nm}$. A FBG deve possuir 50 seções uniformes totalizando um comprimento de 2.5cm. A Tabela 5.6 exhibe o sumário dos resultados obtidos para o GA utilizando decodificação convencional (não suavizada), DSR ($p = \{0.02, 0.05, 0.1\}$) e DSMM ($span = \{2, 5, 10\}$). A decodificação suavizada parece proporcionar melhores resultados do que a decodificação convencional. Particularmente a DSMM parece oferecer melhores resultados do que a DSR. A Fig. 5.9 compara o melhor resultado obtido com suavização DSMM ($span = 5$) com o melhor resultado obtido com a decodificação convencional. A apodização suavizada para λ_B parece seguir um padrão, quase que sempre crescente em função do número da seção, estabilizando-se a partir da seção 20. Por outro lado, a respectiva curva de apodização para o

resultado obtido sem suavização parece seguir um padrão aleatório.

Decodificação e parâmetro	FO			Convergência
	Desvio	Média	Melhor	média
$p = 0.02$	0.4500292	0.2868224	0.1124182	7717.2
Não suavizado	0.0467289	0.1361577	0.0668904	4434.1
$p = 0.1$	0.1310194	0.1032133	0.0413574	4926.7
$span = 2$	0.0697985	0.0958483	0.0303832	4473.9
$span = 5$	0.0574005	0.0511502	0.0219970	4393.5
$p = 0.05$	0.0097613	0.0484910	0.0338847	4642.7
$span = 10$	0.0048111	0.0318944	0.0241869	4240.6

Tabela 5.6: Comparação entre o GA utilizando DSMM para $span = 2, 5$ e 10 ; o GA utilizando DSR para $p = 0.02, 0.05$ e 0.1 ; e o GA não suavizado para o projeto de síntese do filtro triangular.

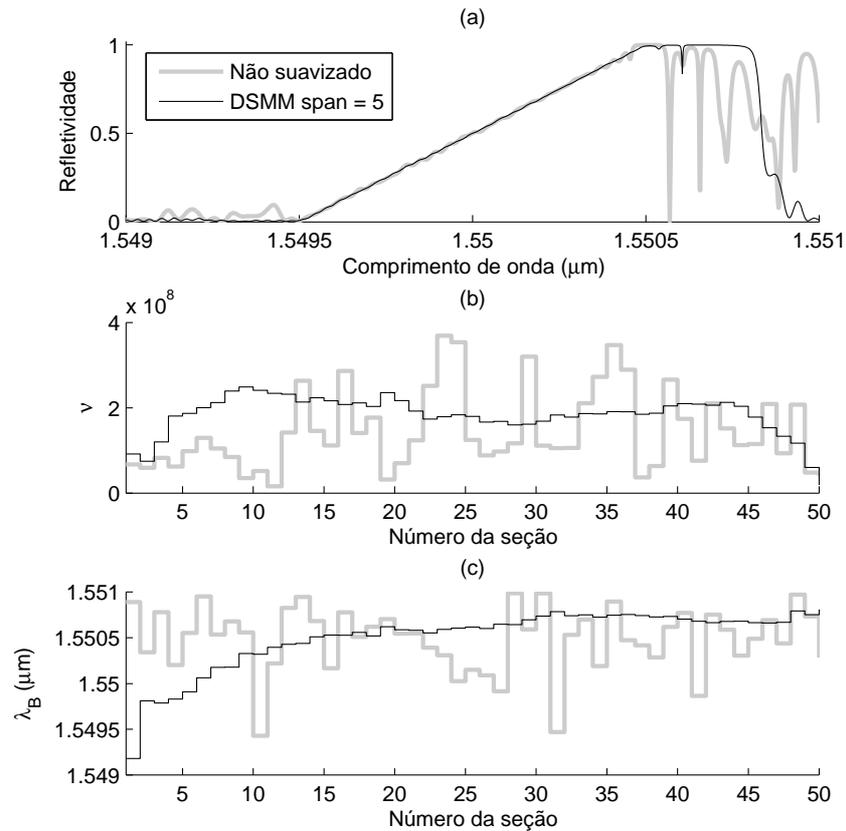


Figura 5.9: Curvas de refletividade (a) e curvas e perfis de apodização para ν (b) e para λ_B (c) de grades obtidas com decodificação convencional e DSMM ($span = 5$).

Uma característica bastante interessante das curvas de apodização suavizadas mostradas na Fig. 5.9 é a simplicidade dos perfis. Este comportamento suscita a possibilidade de atribuir uma curva simples e fixa a ν ou λ_B deixando o GA livre para determinar apenas um parâmetro. Por exemplo, um *chirp* linear seria escolhido, fazendo da curva de λ_B uma reta variando entre os limites 1549nm e 1551nm. Também seria possível fixar uma apodização cosseno-levantado para ν de deixar o GA obter a apodização mais adequada para λ_B .

Evidentemente seria difícil determinar manualmente o ângulo mais interessante para a reta λ_B bem como a amplitude mais adequada para o cosseno-levantado ν . A maneira mais inteligente de determinar estes valores é deixar o próprio GA fazer isso através da DFC. Este método de decodificação, mostrado no Capítulo 4, permite interpretar apenas um gene (um elemento do vetor cromossomo) e a partir deste determinar toda a curva de apodização para um dado parâmetro. Considere o caso em que deseja-se que a curva de λ_B seja uma

reta e que o GA determine a melhor curva de apodização para ν . Para tanto, define-se a família de curvas entre as retas $\lambda_B^{min}(i) = 1549 + (1551 - 1549)(i - 1)/49$ e $\lambda_B^{max}(i) = 1551 + (1549 - 1551)(i - 1)/49$, onde os valores de λ_B são dados em nanômetros e i representa o número da seção uniforme, que varia de 1 a 50 neste exemplo. A curva de apodização seria definida por $\lambda_B(i) = \lambda_B^{min}(i) + [\lambda_B^{max}(i) - \lambda_B^{min}(i)]y$ onde y representa o gene que controlara a inclinação da reta, isto é, qual curva seria escolhida dentre as integrantes da família. A Fig. 5.10 exibe as algumas curvas deste exemplo, para y variando em passos de 0.1 dentro do intervalo de 0 a 1.

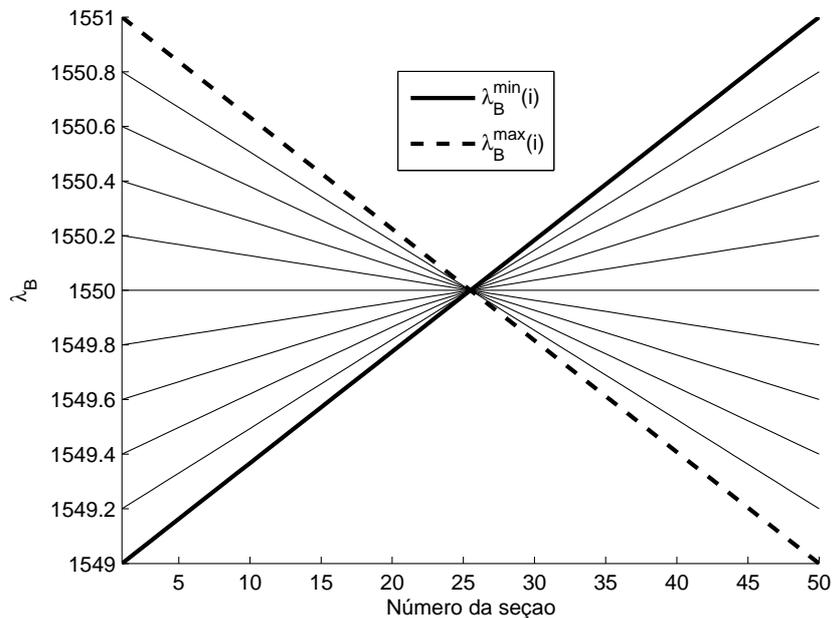


Figura 5.10: Família de retas de possíveis apodizações de λ_B .

Levando a diante este exemplo, a Tabela 5.7 exibe o sumário dos resultados obtidos para o GA utilizando decodificação convencional (não suavizada), DSR para $p = \{0.02, 0.05, 0.1\}$ e DSMM para $span = \{2, 5, 10\}$ para ν . Em todos os casos foram utilizados a DFC para λ_B . Embora o GA não suavizado tenha obtido o melhor resultado (coluna melhor), este também obtém a pior média e o pior desvio. Dentre os que utilizaram suavização para ν , a DSMM com $span = 2$ parece gerar os melhores resultados.

A Fig. 5.11 compara os dois melhores resultados obtidos na Tabela 5.7 para o GA sem DSMM para $span = 2$. As curvas de refletividade bem como as apodizações são bastante similares. É interessante observar que a curva de apodização para ν resulta mais complexa do que para o exemplo anterior onde as duas apodizações, para ν e para λ_B foram determinadas pelo GA. Também os resultados obtidos parecem ser inferiores aos anteriores, o que se expressa

Decodificação e parâmetro	FO			Convergência
	Desvio	Média	Melhor	média
Não suavizado	1.2868522	0.8493544	0.0370774	3280.3
$p = 0.02$	0.0031369	0.2672734	0.2645582	4412.8
$p = 0.05$	0.0107501	0.1563464	0.1418069	6646.9
$span = 10$	0.0536367	0.0967634	0.0681963	7683.2
$p = 0.1$	0.0195136	0.0803373	0.0655212	5372.3
$span = 5$	0.0018217	0.0609929	0.0590037	4354.1
$span = 2$	0.0020842	0.0412569	0.0383233	4875.4

Tabela 5.7: Comparação entre o GA utilizando diferentes esquemas de decodificação suavizadas e uma não suavizada para ν e DFC para λ_B .

também em números quando as Tabelas 5.6 e 5.7 são comparadas.

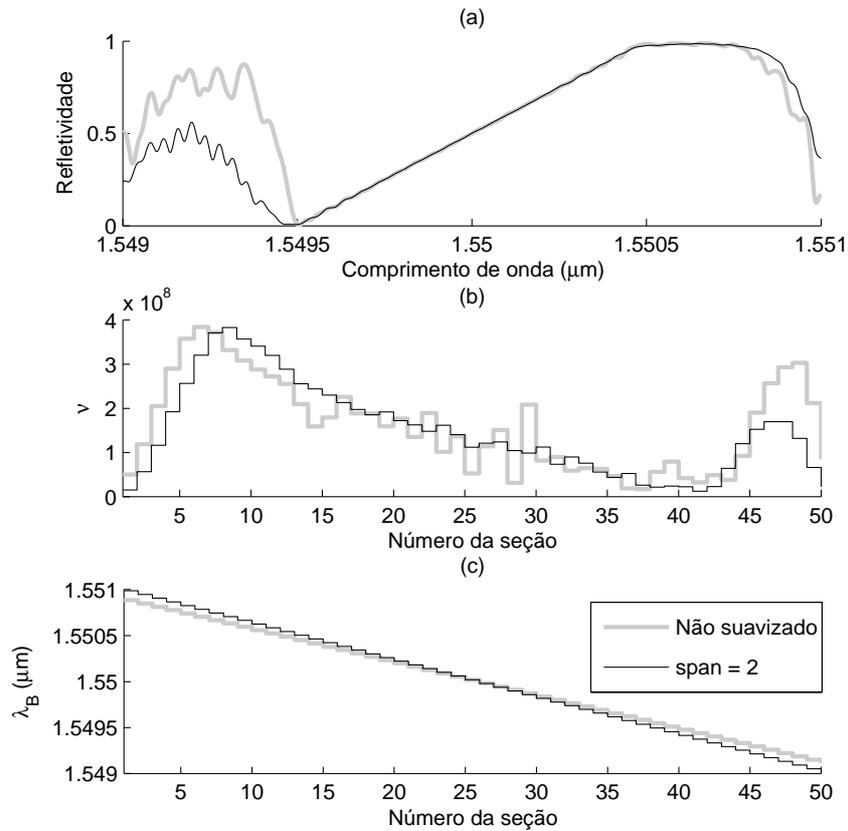


Figura 5.11: Curvas de refletividade (a) e curvas e perfis de apodização para ν (b) e para λ_B (c) de grades obtidas com decodificação convencional e DSMM ($span = 2$) para ν . Ambas utilizando DFC para λ_B .

A Tabela 5.8 resume os resultados para o GA utilizando decodificação suavizada para λ_B (valores entre 1.549nm e 1.551nm) e DFC para ν , utilizando como curva mínima $\nu^{min}(i) = 0$ e como curva máxima um perfil cosseno-levantado de amplitude 4×10^8 . Os melhores resultados foram obtidos utilizando DSMM com $span = 10$. A Fig. 5.12 exibe os melhores resultados sem suavização e com suavização. Nota-se que as soluções obtidas são similares nas curvas de apodização, mesmo para λ_B , onde o formato obtido com a suavização parece uma versão aperfeiçoada e menos caótica do formato obtido sem suavização.

Decodificação e parâmetro	FO			Convergência
	Desvio	Média	Melhor	média
$p = 0.02$	0.9205652	1.2824348	0.4073985	1491.1
$span = 2$	0.1676856	0.2509762	0.0477784	5635.3
$span = 5$	0.2460089	0.1243157	0.0204645	4342.3
Não suavizado	0.0268073	0.1234453	0.0919469	3825.7
$p = 0.05$	0.2766344	0.1183539	0.0216446	3584.6
$p = 0.1$	0.0469346	0.0653660	0.0375708	3997.0
$span = 10$	0.0018530	0.0249884	0.0227560	3598.7

Tabela 5.8: Comparação entre o GA utilizando diferentes esquemas de decodificação suavizadas e uma não suavizada para λ_B e DFC para ν .

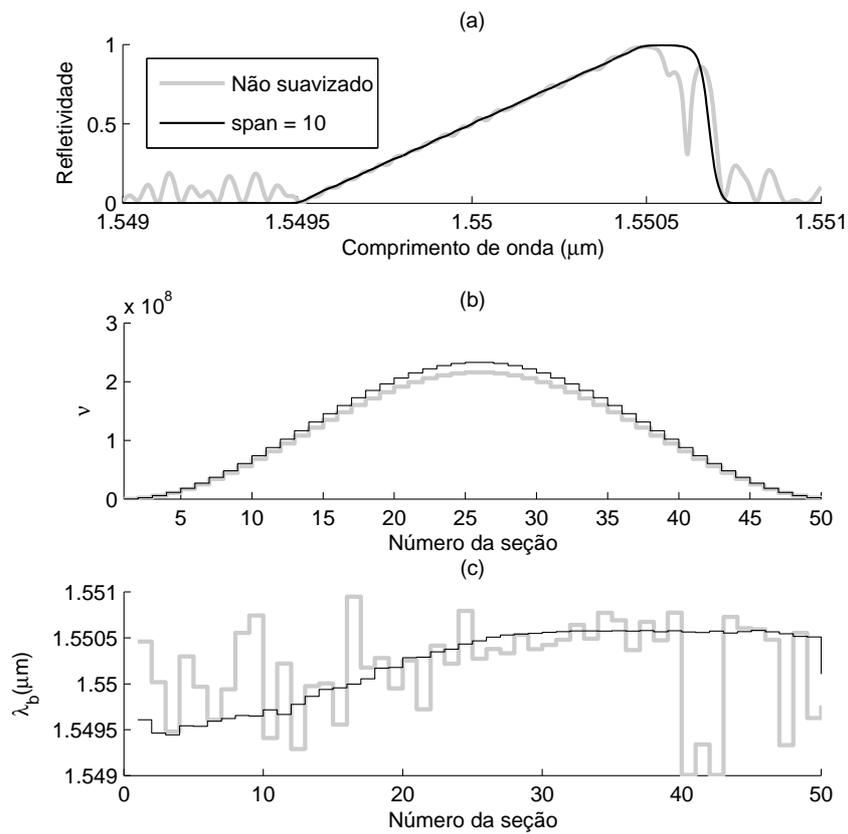


Figura 5.12: Curvas de refletividade (a) e curvas e perfis de apodização para ν (b) e para λ_B (c) de grades obtidas com decodificação convencional e DSMM ($span = 10$) para λ_B . Ambas utilizando DFC para ν .

5.1.3 A decodificação interpolada ou aproximada

A decodificação suavizada claramente demonstra a possibilidade de uma curva relativamente simples representar com sucesso o perfil de apodização ideal de uma FBG. Baseado nesse princípio, seria possível empregar poucos pontos “mestres” e, através destes, definir um perfil completo através de um processo de interpolação ou aproximação. No Capítulo 4 foram consideradas a interpolação linear e as aproximações baseadas nas splines quadráticas e cúbicas. Nesta seção serão testadas estas técnicas de decodificação aplicadas na síntese do filtro refletor simples entre 1548nm e 15502nm e o filtro rampa entre 15495nm e 15505nm, ambos projetos abordados nos itens anteriores.

Seguindo a formulação do Capítulo 4, M_k pontos mestres devem controlar M parâmetros para as M seções uniformes, onde k pode assumir valores 1, 2, 3 e 4, associados aos parâmetros δn_{eff} , ν , Λ (ou λ_B) e δz respectivamente. É desnecessário observar quer, como ϕ (deslocamento discreto de fase) não costuma empregar valores contínuos, tornando-se inútil o uso de qualquer técnica de decodificação baseada em interpolações ou quaisquer tipos de suavizações para este parâmetro, pelo menos para a classe de projetos abordada nesta Tese.

Como M_k costuma ser de 5 a 10 vezes menor do que M , a decodificação interpolada implica em uma severa redução da quantidade de parâmetros ou dimensões do problema. Portanto, espera-se que o GA utilizando este esquema necessite populações menores ou permita convergências em muito menos gerações quando utilizando os mesmos tamanhos de população empregados anteriormente, isto é, 40 indivíduos descendentes. Considere o projeto de refletor simples entre 1549.8nm e 1550.2nm, com otimização da apodização de ν utilizando decodificação interpolada com 7 pontos mestres. Esta quantidade parece ser suficiente para descrever uma curva de apodização como aquela mostradas na Fig. 5.8. Tomando como verdadeira esta suposição, a Tabela 5.9 mostra o sumário de várias simulações utilizando relações de 2/5, 2/10, 4/20 e 8/40 de indivíduos progenitores / indivíduos descendentes (foi empregada a seleção elitista). O restante dos parâmetros foram os mesmos dos exemplos anteriores, exceto pelo valor de $\tau_M = 0.1$ para todas as simulações.

A Tabela 5.9 mostra que os resultados são todos bastante similares. Porém, percebe-se facilmente que a DSC parece obter resultados sempre melhores do que DSQ, que por sua vez obtém melhores resultados do que a DIL. Também é fácil notar que os resultados são tanto melhores quanto maior o número de indivíduos. Seria simples recomendar sempre o uso da maior quantidade de indivíduos e sempre a DSC. Porém o número de chamadas da FO é proporcional ao número de descendentes por geração, fazendo com que a convergência do GA com 20 indivíduos seja quase 2 vezes mais rápida do que a convergência para o GA com 40 indivíduos tomando como base o mesmo número de gerações. O número total de chamadas

Decodificação	Progenitores/ Descendentes	FO			Convergência
		Desvio	Média	Melhor	média
DIL	2/5	0.0000221	0.0286154	0.0285915	372.3
DSQ	2/5	0.0000417	0.0227399	0.0226946	773.4
DSC	2/5	0.0001999	0.0203604	0.0201192	751.0
DIL	2/10	0.0000027	0.0285806	0.0285753	198.4
DSQ	2/10	0.0000023	0.0226683	0.0226640	259.0
DSC	2/10	0.0000137	0.0199863	0.0199653	376.6
DIL	4/20	0.0000021	0.0285775	0.0285748	138.7
DSQ	4/20	0.0000025	0.0226641	0.0226612	197.9
DSC	4/20	0.0000059	0.0199698	0.0199614	244.1
DIL	8/40	0.0000016	0.0285755	0.0285732	91.4
DSQ	8/40	0.0000024	0.0226594	0.0226565	193.2
DSC	8/40	0.0000025	0.0199623	0.0199586	218.1

Tabela 5.9: Comparação entre o GA utilizando DIL, DSQ e DSC para três diferentes combinações de números de progenitores e descendentes.

à função objetivo é similar para populações de 5, 10 e 20 indivíduos. Entretanto, o melhor custo benefício (relativo ao tempo de convergência esperado e ao valor FO médio) parece ser obtido para uma população de 10 descendentes empregando DSC.

A Fig. 5.13 compara os espectros e as apodizações das grades obtidas pelo GA empregando as 3 estratégias de decodificação suavizadas. Os três espectros de refletividade são bastante similares bem como as respectivas curvas de apodização; muito embora a curva de apodização obtida com a suavização linear resulte refletida em relação ao eixo das abscissas quando comparada às demais.

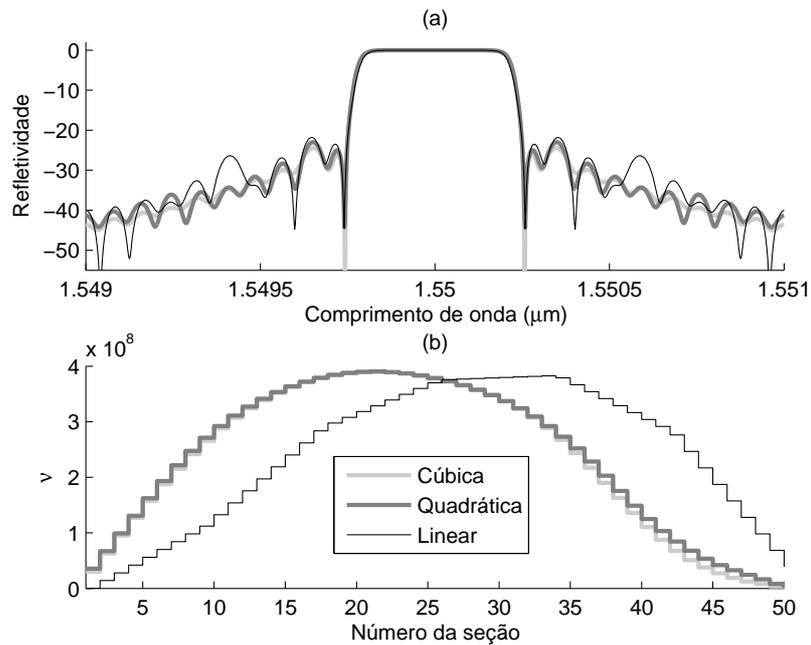


Figura 5.13: Curvas de refletividade (a) e curvas e perfis de apodização para ν (b) de grades obtidas com decodificação suavizada cúbica, quadrática e linear.

Comparando a Tabela 5.9 e a Tabela 5.5 observa-se que a técnica de decodificação interpolada ou aproximada é realmente capaz de melhorar a convergência média por um fator de aproximadamente 10. Bons resultados são também obtidos com populações reduzidas, com um número de descendentes por geração entre 5 e 10. Porém é preciso lembrar que a redução da dimensionalidade do problema pode afetar também o valor ideal para o parâmetro da mutação auto-adaptativa (τ_M). Para investigar essa possibilidade, o GA empregando população de 20 indivíduos e codificação *spline* cúbica com $M_k = 5, 6, 7, 8, 9, 10$ e 14 sobre ν , foi aplicado dez vezes para 18 valores diferentes de τ_M entre 0.01 e 0.9. Para cada grupo de dez aplicações foi coletado o valor final médio para a FO final. Os melhores resultados médios da FO juntamente com o valor mais bem sucedido de τ_M em função do número de dimensões do problema (M_k) são mostrados na Fig. 5.14. É interessante ressaltar que o melhor resultado médio para a FO cai em função da dimensionalidade, aparentemente estabilizando-se a partir de 10 dimensões. Também é interessante observar como o valor de τ_M diminui em função do número de dimensões. Considerando que τ_M seja proporcional a $1/\sqrt{D}$, a expressão:

$$\tau_M \approx 0.0865/\sqrt{D}, \quad (5.3)$$

cuja curva é mostrada em Fig. 5.14, seria uma aproximação provavelmente mais confiável do que aquela (5.2), pelo menos para este caso específico empregando decodificação *spline*.

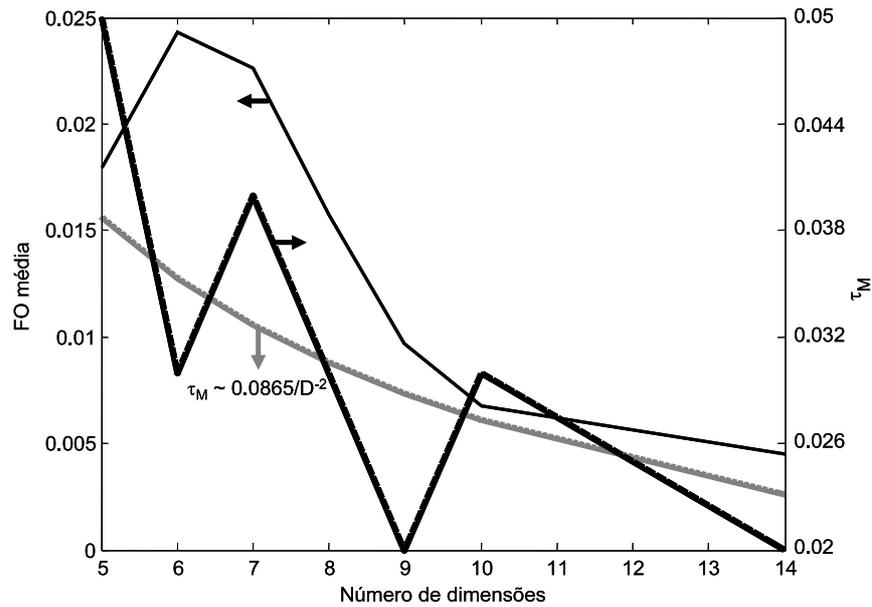


Figura 5.14: Curvas de refletividade (a) e curvas e perfis de apodização para ν (b) de grades obtidas com DSC, DSQ e DIL.

5.1.3.1 Decodificação *spline* para a síntese de filtros triangulares

Similarmente como feito para a decodificação suavizada, projetos de filtros triangulares também podem ser sintetizados empregando *chirp* e decodificação *spline*. É possível deixar com que o GA otimize o perfil de δn_{eff} ou de ν , com um número arbitrário de genes, empregando a decodificação *spline* enquanto o perfil de λ_B ou Λ seja otimizado com apenas um gene, através da decodificação DFC (usando uma família de retas). Porém o inverso também é possível, isto é, deixar com que o GA otimize o perfil de λ_B através da decodificação *spline* e o perfil de ν ou δn_{eff} através da decodificação por família de curvas (família de cosseno-levantados). Existem também outras possibilidades interessantes, como por exemplo empregar a decodificação *spline* em ambos ν e λ_B , ou combinar a decodificação por interpolação linear de dois pontos para λ_B com a decodificação *spline* para δn_{eff} .

Para investigar a decodificação *spline* para a síntese de filtros triangulares, foram testadas várias combinações de parâmetros ou estratégias diferentes cujos resultados são sumarizados na Tabela 5.10. Foram considerados 5 tipos de combinações:

- “A”: ambos, δn_{eff} ou ν , e λ_B são obtidos através DSQ ou DSC com 5 ou 7 pontos mestres. O número total de parâmetros (ou genes) pode ser 10 ou 14.
- “B”: apenas δn_{eff} ou ν são obtidos por DSQ ou DSC com 5 ou 7 pontos mestres. O

parâmetro λ_B é obtido através de DIL com dois pontos (mais dois genes). O número total de genes pode ser 7 ou 9.

- “C”: apenas δn_{eff} ou ν são obtidos por DSQ ou DSC com 5 ou 7 pontos mestres. O parâmetro λ_B é obtido através de DFC, onde apenas um gene controla a inclinação de uma reta que define o *chirp* linear, tal como mostrado na Fig. 5.10. O número total de genes pode ser 6 ou 8.
- “D”: apenas δn_{eff} ou ν são obtidos por DSQ ou DSC com 5 ou 7 pontos mestres. O *chirp* é fixo fazendo λ_B variar de $1.549\mu\text{m}$ até $1.551\mu\text{m}$ ao longo do comprimento da grade. O número total de genes pode ser 5 ou 7.
- “E”: δn_{eff} ou ν são obtidos através da DFC com uma família do tipo cosseno-levantado com amplitude máxima de 4×10^{-4} e mínima de 0. O parâmetro λ_B é obtido através da DSQ ou DSC com 5 ou 7 pontos mestres. O número total de genes pode ser 6 ou 8.

Comb.	Num.	Decod.			FO			Conv.
		δn_{eff}	ν	λ_B	Desvio	Média	Melhor	média
A	1		DSC7	DSC7	0.174966	0.174733	0.038322	2968.1
	2		DSC5	DSC5	0.228173	0.173556	0.078773	1922.6
	3	DSC5		DSC5	0.165710	0.152081	0.059544	1787.6
B	1	DSQ7		DIL2	0.002795	0.087212	0.079482	1415.9
	2		DSC7	DIL2	0.027577	0.086276	0.064782	1290.8
	3	DSQ7		DIL2	0.024430	0.072545	0.057019	1540.9
C	1	DSC7		DFC	0.007786	0.088849	0.077541	969.3
	2	DSC5		DFC	0.000011	0.083880	0.083872	484.8
	3	DSQ5		DFC	0.000014	0.072528	0.072505	572.2
D	1	DSC7			0.000066	0.180130	0.180037	778.5
	2		DSC7		0.000033	0.144468	0.144399	266.1
	3		DSQ7		0.000043	0.137160	0.137110	298.1
E	1	DFC		DSC5	0.000506	0.060846	0.060659	854.3
	2	DFC		DSC7	0.008674	0.056163	0.040159	1061.6
	3		DFC	DSC7	0.009930	0.047742	0.034205	1439.9

Tabela 5.10: Resumo dos resultados obtidos, listando os três melhores resultados para cada tipo de combinação.

A coluna “Num.” (número) identifica a combinação para cada um dos três resultados selecionados para cada classe de combinação. A coluna “Decod.” (decodificação) exibe o tipo de decodificação utilizado para cada um dos parâmetros relacionados nas três subcolunas δn_{eff} , ν e λ_B . As siglas DSQ, DSC, DFC, e DIL precedem números que indicam o quantidade de pontos mestres.

A Fig. 5.15 exibe as curvas típicas de refletividade em função do comprimento de onda; as curvas de “amplitude” da perturbação do índice de refração, isto é, o produto $\nu \times \delta n_{eff}$; e as curvas de λ_B em função do número da seção para a classe de combinação “A”. As figuras Fig. 5.16, Fig. 5.17, Fig. 5.18 e Fig. 5.19 exibem um conteúdo similar respectivamente para as classes de combinação “B”, “C”, “D” e “E”.

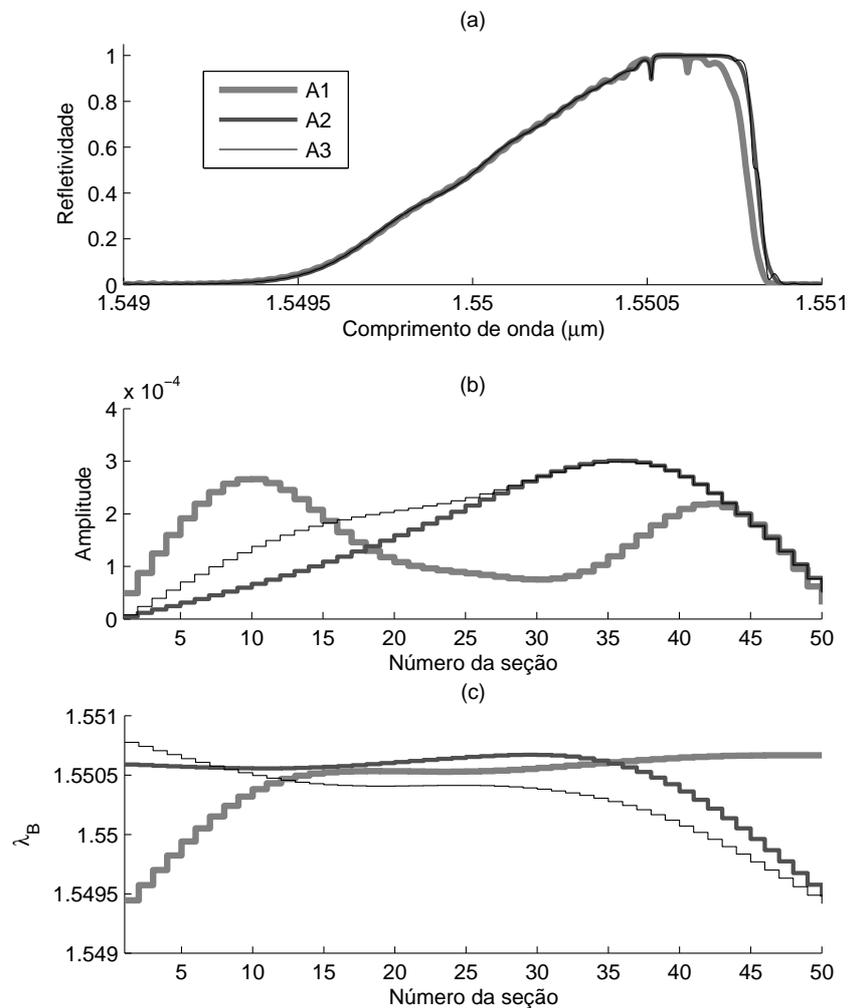


Figura 5.15: Curvas de refletividade (a) e curvas de amplitude de perturbação do índice (b) e o perfil de λ_B (c).

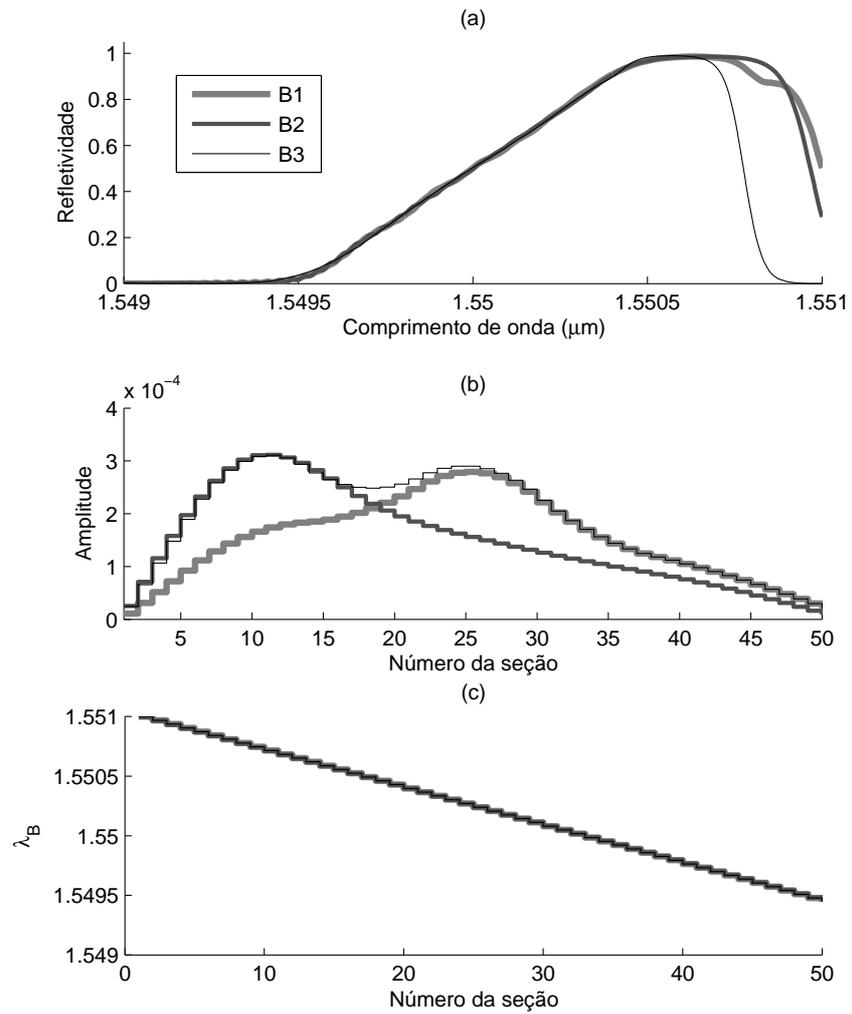


Figura 5.16: Curvas de refletividade (a) e curvas de amplitude de perturbação do índice (b) e o perfil de λ_B (c).

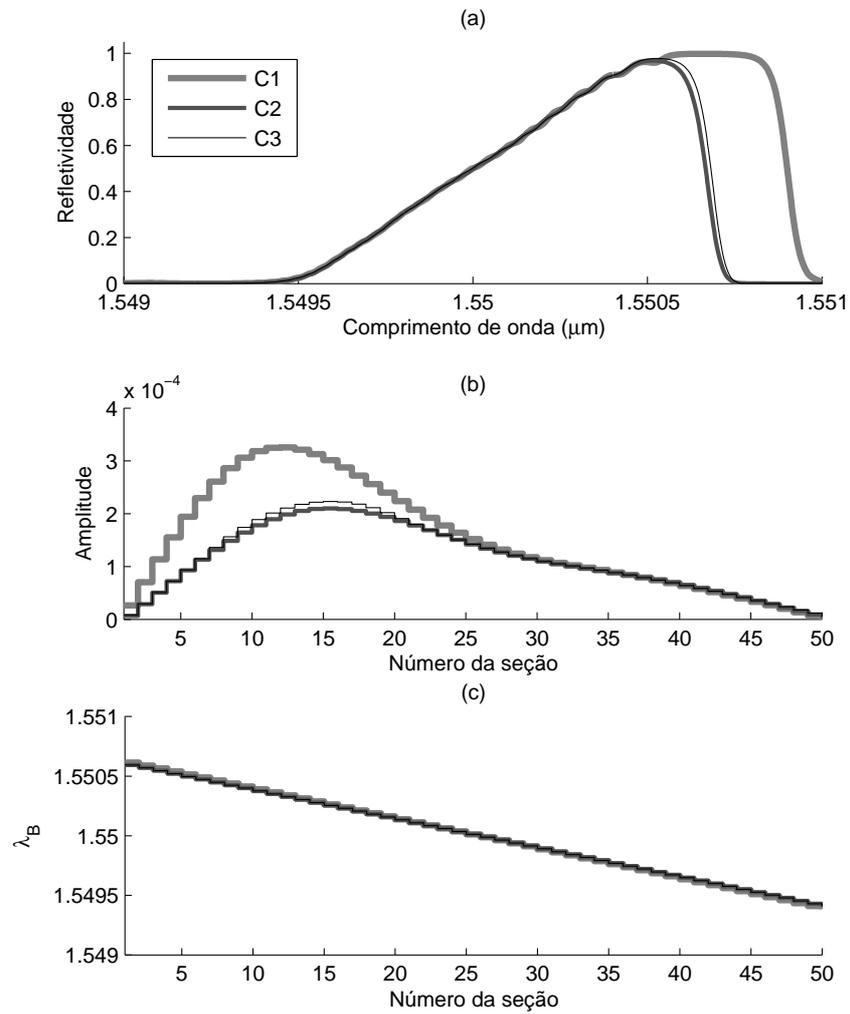


Figura 5.17: Curvas de refletividade (a) e curvas de amplitude de perturbação do índice (b) e o perfil de λ_B (c).

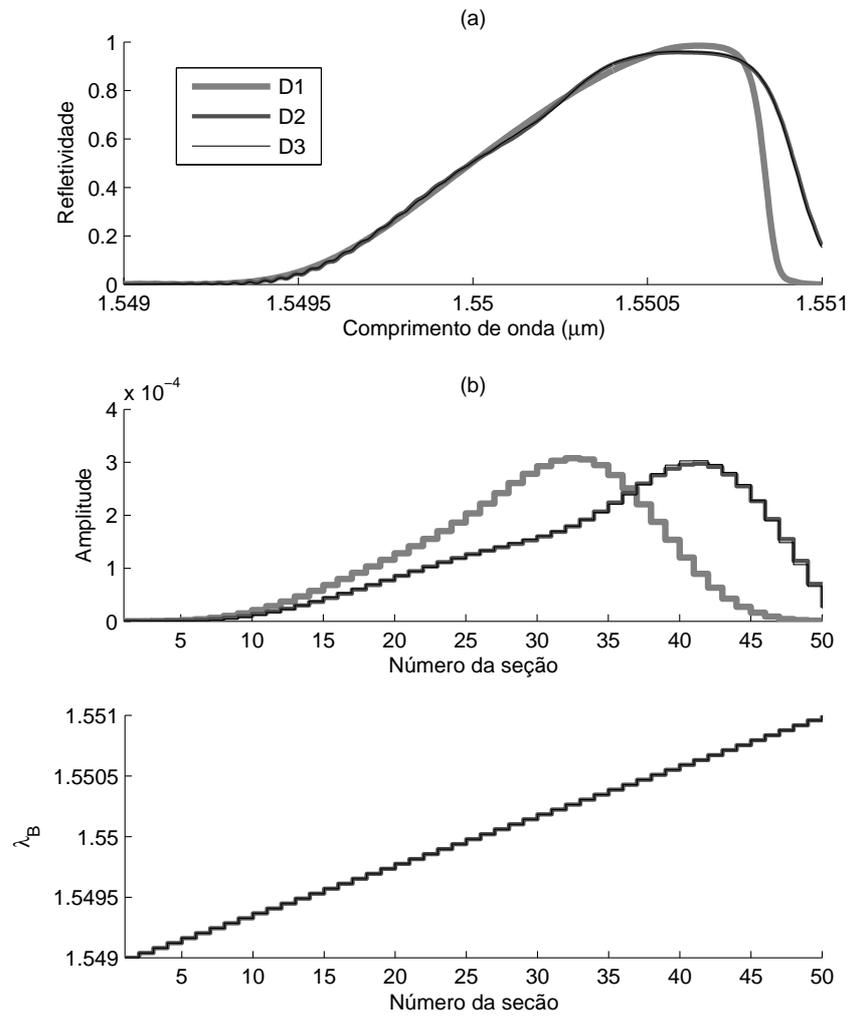


Figura 5.18: Curvas de refletividade (a) e curvas de amplitude de perturbação do índice (b) e o perfil de λ_B (c).

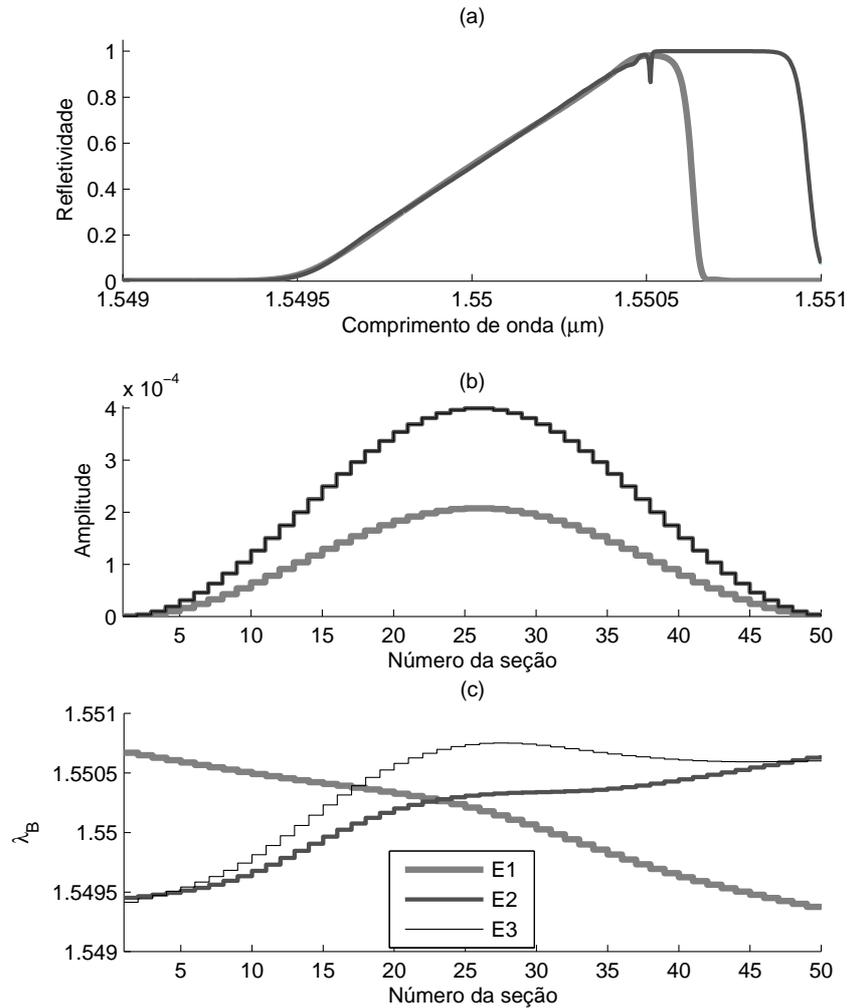


Figura 5.19: Curvas de refletividade (a) e curvas de amplitude de perturbação do índice (b) e o perfil de λ_B (c).

De uma forma geral, a decodificação *spline* cúbica com 7 pontos mestres parece oferecer melhores resultados do que a *spline* quadrática ou o uso de 5 pontos mestres na maioria dos projetos. Quando se considera o mesmo número de pontos mestres, a vantagem também é da *spline* cúbica, embora esta seja freqüentemente muito pequena. Por exemplo, os projetos C2, C3 resultaram muito parecidos, assim como os D2 e D3.

Um fato que também merece ser comentado é a ocorrência de mais projetos bem sucedidos através otimização do parâmetro δn_{eff} no lugar de ν . Isto se deve pelo fato do comprimento de onda de máxima refletividade ser dado também em função δn_{eff} . Portanto, além do controle da refletividade, este parâmetro oferece ao GA o poder de produzir deslocamentos

espectrais que manifestam-se na forma de um alargamento espectral similar ao obtido pelo uso de *chirp*, embora com intensidade bastante limitada. Em outras palavras, a otimização de δn_{eff} oferece mais graus de liberdade ao GA. Curiosamente, a aplicação da decodificação *spline* sobre λ_B e decodificação por família de curvas sobre δn_{eff} ou ν parece ser a melhor de todas as estratégias, sendo “E”, na Tabela 5.10, claramente a combinação mais bem sucedida.

5.1.4 Aceleração do GA por meio da redução da quantidade de amostras da FO

Todos os exemplos mostrados neste capítulo empregaram alguma FO baseada no cálculo do erro RMS entre a curva alvo e a curva de refletividade calculada para 100 amostras fixas. Obviamente este número de amostras pode ser modificado com importantes impactos sobre o desempenho do GA, sendo interessante investigar o efeito deste parâmetro. Também é objetivo desta seção determinar o quanto a técnica de aceleração (GA acelerado) apresentado no Capítulo 4 pode ser melhor do que a o GA original sem aceleração em termos de desempenho computacional ou em relação à qualidade dos filtros sintetizados.

Considere dois projetos, um filtro refletor simples de 1549.8nm a 1550.2nm e um filtro rampa de 1549.5nm a 1550.5nm, ambos largamente abordados nas seções anteriores. Considere a codificação de ν em ambos com um *chirp* linear definido através da decodificação por família de curvas especificamente para o projeto rampa. O número de pontos mestres igual a 7 e tamanho de população de 20 descendentes (com 5 progenitores por geração). Para o projeto do refletor simples são ao todo 7 parâmetros, enquanto que para o projeto rampa são 8 parâmetros. Para estas dimensionalidades, de acordo com (5.3), os valores mais apropriados para τ_M seriam respectivamente 0.0327 e 0.0306. Por simplicidade e por tratar-se de valores relativamente próximos, foi adotado $\tau_M = 0.03$.

A Fig. 5.20 exhibe valores de erro RMS médios e o desvio padrão (através das barras) calculados entre as curvas alvo e as curvas de refletividade de 10 filtros FBG. Para viabilizar comparações, os valores de erro RMS foram obtidos utilizando 1000 amostras fixas e homogeneamente espaçadas. Os valores de abcissa são o número de amostras utilizadas no cálculo da FO pelo GA ao longo das suas iterações. Na Fig. 5.20 (a) estão as curvas associadas ao projeto do filtro refletor simples; em (b) as curvas para o projeto de filtro rampa. Em ambos (a) e (b) é evidente que as curvas de erro apenas se estabilizam a partir de 100 amostras. Não por coincidência este foi o valor de amostras adotado como padrão nas seções anteriores deste capítulo. As curvas rotuladas como “A” foram obtidas por um GA utilizando amostras homogeneamente espaçadas e sempre posicionadas nos mesmos comprimentos de onda ao longo das

iterações. As curvas rotuladas como “B” foram obtidas por um GA utilizando amostragem AAL. As curvas rotuladas como “C” foram obtidas por um GA “acelerado”, tal como proposto na seção 4.3.4.1, isto é, com o uso do procedimento de correção elitista com $S_S = 100$, $N_E = 1$ além do uso de amostragem AAL. É interessante observar que as curvas “C” atingem valores relativamente baixos de erro RMS mesmo quando apenas uma única amostra é utilizada. De fato, o comportamento do erro RMS da curva “C” é similar ao comportamento da curva “A” com 100 amostras para todos os valores de S_R , analisados, desde 1 até 50.

A Fig. 5.21 exibe os espectros de refletividade para os dois projetos, o refletor simples e o filtro rampa, todos obtidos utilizando apenas uma única amostra para o cálculo da FO. Os códigos “A”, “B” e “C” empregados nesta figura possuem os mesmos significados dos códigos da Fig.5.20. É indiscutível a superioridade dos resultados obtidos com o GA acelerado. Particularmente para o projeto do filtro refletor simples, o resultado obtido pelo GA acelerado é bastante satisfatório.

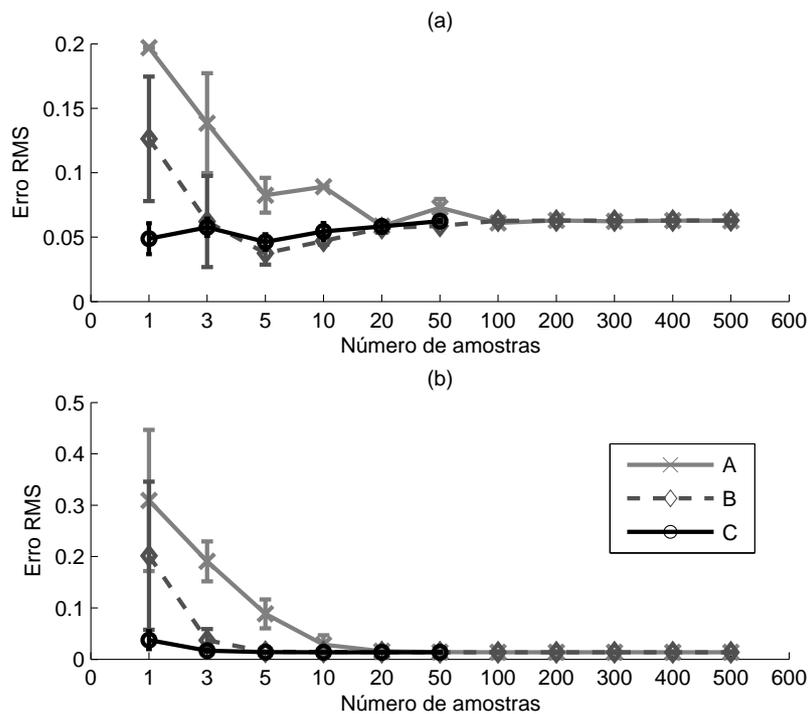


Figura 5.20: Curvas de erro RMS em função do número de amostras (empregado pela FO) para o GA com amostragem fixa, aleatoriamente deslocada e para o GA acelerado: (a) curvas obtidas para o projeto refletor simples; (b) curvas obtidas para o projeto filtro rampa.

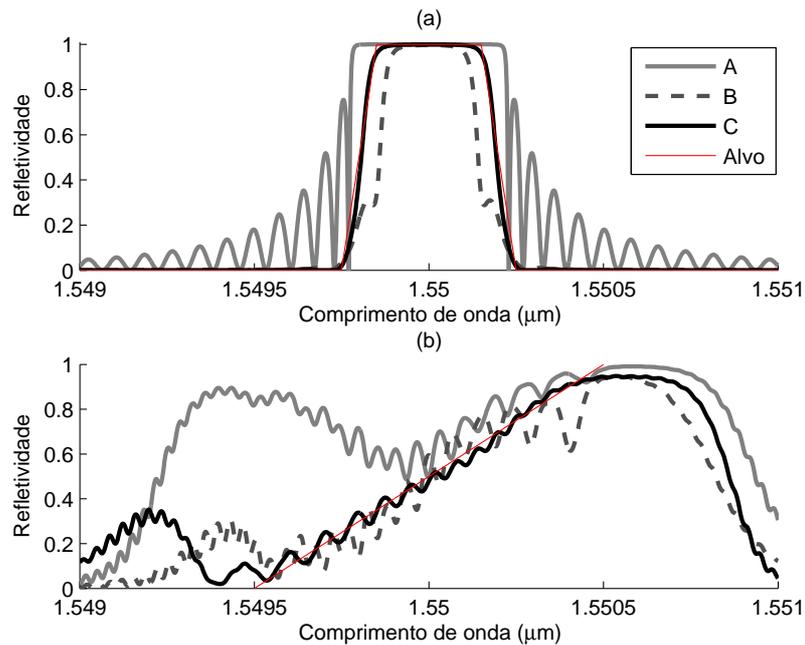


Figura 5.21: Curvas de refletividade obtidas para o GA empregando FO com apenas 1 amostra ($S_S = 100$ para o GA acelerado “C”): (a) curvas de refletividade para o projeto refletor simples; (b) curvas de refletividade para o projeto filtro rampa.

A Fig. 5.22 exhibe curvas de ganho calculadas a partir da relação entre o tempo de processamento para o GA não acelerado com 100 amostras e o GA acelerado com $S_S = 100$ para diversos valores de S_R . A curva Fig. 5.22 (b) exhibe o ganho calculado a partir do tempo de processamento médio de uma geração, enquanto Fig. 5.22 (a) exhibe o ganho calculado a partir do tempo de processamento médio total do GA. A curva “ideal” mostrada em (a) e (b) foi obtida a partir de (4.26). As curvas obtidas para Fig. 5.22 (a) para ambos os projetos resultaram bastante próximas à ideal para valores de $S_R > 5$. Este comportamento pode ser explicado pelo fato de (4.26) ignorar o custo computacional do GA, considerando-o desprezível em comparação ao custo acumulado associado ao processamento de todas as amostras das curvas de refletividade da população de filtros de uma geração qualquer. Quando considerado o tempo acumulado de todas as gerações, o resultado é ainda mais distante da curva ideal, tal como pode ser avaliado através da Fig. 5.22 (b). Por exemplo, o GA acelerado apresenta ganhos insignificantes quando $S_R > 10$ para o projeto refletor simples, porém obtém ganhos superiores ao ideal para $S_R = 5$. Por outro lado, para o projeto filtro rampa, o ganho é praticamente constante para S_R entre 5 e 20. Em outras palavras, embora 5 amostras possibilitem um ganho de aproximadamente 9 vezes considerando o tempo de processamento por geração, mais gerações acabam sendo necessárias de modo que os tempos totais médios

sejam praticamente os mesmos para $S_R = 5$ até $S_R = 20$. Conclui-se, portanto, que a aplicação do GA acelerado implica em um número maior de chamadas da FO, na prática permitindo ganhos em torno de 5 vezes. O valor ideal para S_R (para $S_S = 100$) seria algo entre 5 e 10.

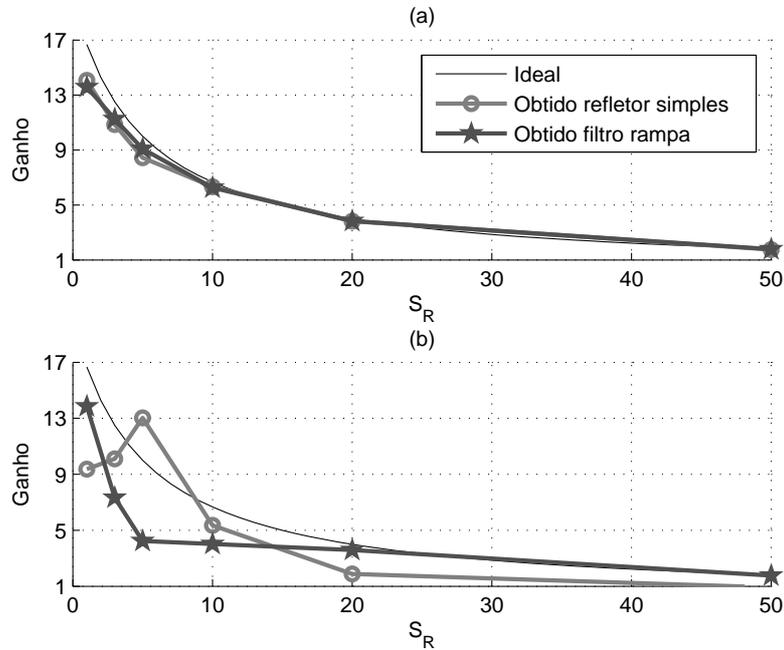


Figura 5.22: Curvas de ganho calculadas em função do tempo de processamento por geração (a) e em função do tempo de processamento total do GA.

5.2 PSO: estudo das técnicas de decodificação e de aceleração

Esta seção tem o objetivo de apresentar o PSO e o EPSO aplicados ao problema de síntese de FBGs da mesma forma como foi feito para o GA na seção anterior. Portanto, primeiramente serão investigados alguns parâmetros considerados básicos, como o número ideal de focos de busca, isto é, o tamanho ideal de enxame, o efeito de técnicas adaptativas e auto-adaptativas para a regulação dos pesos e inércia ao longo do processo de otimização. Em seguida serão investigados os efeitos das técnicas de decodificação e de aceleração apresentadas no Capítulo 4.

5.2.1 Parâmetros básicos do PSO

5.2.1.1 Tamanho do enxame

Assim como para o GA, um dos parâmetros fundamentais do PSO é o número de focos de busca ou, em outras palavras, o número de partículas utilizadas no enxame (considerando que este número não varie ao longo das iterações). Para avaliar o valor mais apropriado para este parâmetro, serão utilizados os mesmos projetos das seções anteriores: o filtro refletor simples e o filtro rampa.

Considere inicialmente a decodificação convencional com 50 seções. Para o projeto de filtro refletor simples com codificação de ν resultam 50 parâmetros ou 50 dimensões. Para o projeto de filtro rampa, 50 seções implicam em 100 dimensões considerando que δn_{eff} e λ_B sejam codificados. A Tabela exibe o resumo dos resultados coletados para 5, 10, 20, 40 e 50 partículas. Os pesos utilizados foram fixos e iguais a 0.729, 2.0412 e 0.9477, respectivamente para a inércia, memória e cooperação. Tais valores foram retirados da literatura [43]. Para cada tamanho de enxame o PSO foi corrido 10 vezes. A Fig. 5.23 exibe o valor médio de FO em função do número de partículas no enxame para os dois projetos. É interessante observar como a curva parece convergir a partir de 60 indivíduos no enxame para ambos os projetos. Assim como para o GA em relação ao tamanho da população, o PSO também parece exigir um número de partículas de ordem equivalente ao número de dimensões do problema.

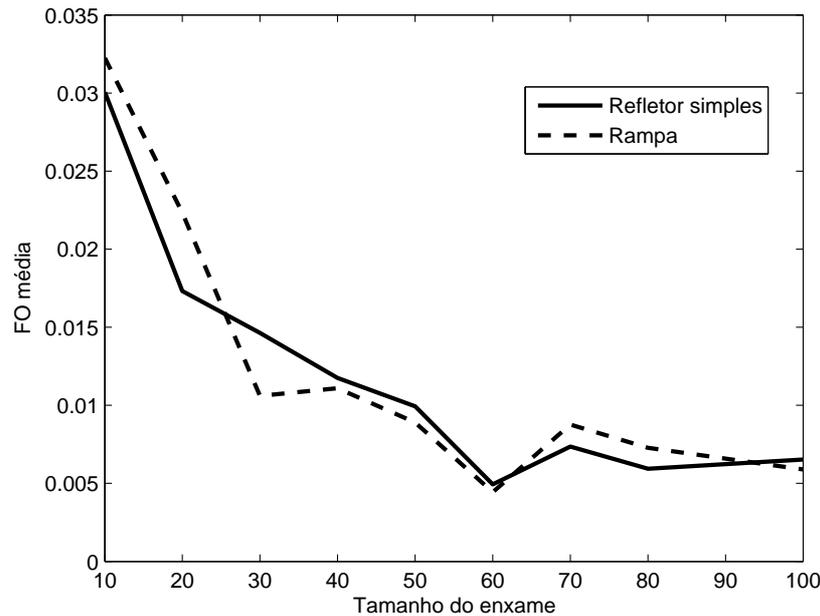


Figura 5.23: Curva de *speed-up* efetivo em função de S_R para o GA acelerado em relação ao convencional utilizando 100 amostras fixas.

5.2.1.2 Pesos fixos, dinâmicos e auto-adaptativos

Assim como ocorre com os GA, técnicas dinâmicas e auto-adaptativas também podem ter um bom impacto do desempenho do PSO. De acordo com a literatura, técnicas dinâmicas tendem a oferecer melhores resultados do que a escolha de pesos fixos [43]. Também técnicas auto-adaptativas tendem a ser melhores do que técnicas dinâmicas [47].

Considere os mesmos projetos abordados na seção anterior, o filtro refletor simples e o filtro rampa. Para estes dois projetos o PSO utilizando pesos fixos, o PSO utilizando pesos dinâmicos e o EPSO foram aplicados para uma enxame de 30 e de 60 indivíduos. Os pesos fixos foram os mesmos utilizados na seção anterior. Para o PSO com pesos dinâmicos, apenas a inércia foi atualizada linearmente em função do número de iterações, decrescendo de 0.9 até 0.4 em 10000 iterações. Os outros pesos, relativos à memória e à cooperação, foram fixos em 1.49 [43]. Para o EPSO com $N_\lambda = 2$, os pesos foram atualizados através de mutações lognormais com desvios de 0.1. A inércia e o ruído da partícula global c_3 foram mantidos dentro do intervalo $[0,1]$, enquanto que os demais pesos foram mantidos no intervalo $[0,2]$.

PSO	Filtro	Tamanho do enxame	FO			Convergência
			Desvio	Média	Melhor	Média
Estático	Refletor	30	0.00767	0.02225	0.01356	1537.3
Dinâmico	Refletor	30	0.01196	0.02026	0.00678	5274.9
Evolucionário	Refletor	30	0.00009	0.00312	0.00297	7579.1
Estático	Rampa	30	0.12668	0.35170	0.21881	3915.5
Dinâmico	Rampa	30	0.18837	0.45669	0.25033	6211.5
Evolucionário	Rampa	30	0.08825	0.28080	0.11337	8847.7
Estático	Refletor	60	0.00722	0.01555	0.00677	3230.4
Dinâmico	Refletor	60	0.00866	0.01840	0.00948	3909.8
Evolucionário	Refletor	60	0.00004	0.00308	0.00301	6763.1
Estático	Rampa	60	0.12576	0.24848	0.10474	3458.4
Dinâmico	Rampa	60	0.17252	0.34603	0.16501	5236.5
Evolucionário	Rampa	60	0.08151	0.16820	0.07579	9666.2

Tabela 5.11: Comparação entre resultados obtidos para PSO com pesos fixos (PSO estático), PSO com inércia dinâmica (PSO dinâmico) e PSO auto-adaptativo utilizando a estratégia evolucionária do EPSO (PSO evolucionário).

A tabela 5.11 sintetiza os resultados obtidos mostrando um desempenho do PSO estático superior ao observado para o PSO dinâmico. Este comportamento conflitante com as expectativas está provavelmente relacionado à combinação imperfeita de parâmetros, visto que todos eles foram extraídos da literatura onde eram aplicados a problemas completamente diferentes.

O PSO evolucionário (EPSO) obteve os melhores e mais estáveis resultados, tal como esperado. Comparando os resultados do EPSO para o filtro refletor simples observa-se que o aumento do tamanho do enxame não oferece um impacto relevante nos resultados. Para o filtro rampa, por outro lado, o aumento do enxame gera um efeito muito mais evidente. Certamente este comportamento deve-se ao fato do projeto rampa utilizar o dobro de parâmetros, necessitando obviamente de uma população maior e/ou um número maior de gerações para a convergência. De acordo com a Fig. 5.23, o PSO estático necessita de um enxame de tamanho da ordem de 60 indivíduos quando 50 parâmetros são necessários para se descrever uma solução, porém o EPSO parece oferecer melhores resultados com apenas 30 indivíduos.

Seria possível concluir, a partir da Tabela 5.11 e da Fig. 5.23, que o EPSO seria mais eficiente no aproveitamento do enxame. Porém, é preciso lembrar que o EPSO aplica, a cada iteração e para cada partícula, uma pequena estratégia evolutiva de apenas uma geração. Para o EPSO utilizado na Tabela 5.11, cada partícula do enxame gera duas outras partículas levemente modificadas pela mutação. Na prática, quando o EPSO emprega 30 indivíduos no enxame, 90 partículas são avaliadas através da FO a cada iteração. Portanto, o EPSO gera melhores resultados a preço de um custo computacional adicional.

De qualquer forma o EPSO obtém excelentes resultados talvez inacessíveis ao PSO simples. Portanto, este será a variante do PSO utilizada a partir deste ponto neste trabalho. Entretanto, para completar seu estudo preliminar ainda resta avaliar os valores mais apropriados para o parâmetro de mutação τ_P e o parâmetro de cooperação probabilística (associado à estrela estocástica) P_Z .

5.2.1.3 Definição dos parâmetros τ_P e P_Z

Para a definição do parâmetro de mutação τ_P e do parâmetro de cooperação P_Z , os mesmos projetos empregados nas seções anteriores serão reaproveitados. Porém, para reduzir a quantidade de parâmetros utilizados pelo projeto rampa a um valor similar ao utilizado no projeto refletor simples, nesta seção será empregada a decodificação de λ_B baseada na família de curvas definida por duas retas: $\lambda'_B(k) = 1549 + 2 \times (k - 1)/49$ e $\lambda''_B(k) = 1551 - 2 \times (k - 1)/49$. Dessa forma, para o projeto rampa serão empregados 51 parâmetros, deixando ambos os projetos, rampa e filtro refletor simples, com praticamente a mesma dimensionalidade, permitindo dessa forma comparações e análises mais próximas.

Para ambos os projetos, o EPSO foi corrido 5 vezes para cada combinação de $\tau_P = 0.05, 0.1, 0.2, 0.3, 0.5$ e $P_Z = 0, 0.25, 0.5, 0.75, 1$. Os resultados, os valores médios de FO, são mostrados na Tabela 5.12 e 5.13, respectivamente para o projeto do refletor simples e filtro rampa. Para o filtro refletor, o melhor resultado foi obtido para $\tau_P = 0.1$ e $P_Z = 0.5$. Para o filtro rampa, o melhor resultado foi obtido para $\tau_P = 0.05$ e $P_Z = 0.25$. Estas combinações de parâmetros foram o ponto de partida para outros testes cujos resultados são mostrados nas Fig. 5.24 e 5.25. A Fig. 5.24(a) explora outros valores de P_Z no intervalo $[0.25, 0.75]$, conservando $\tau_P = 0.1$. A Fig. 5.24(b) explora outros valores de τ_P para o $[0.05, 0.2]$, conservando $P_Z = 0.5$. A Fig. 5.25(a) faz o mesmo que 5.24(a), porém para P_Z no intervalo $[0, 0.5]$ e $\tau_P = 0.05$. A Fig. 5.25(b) faz o mesmo que 5.24(b), porém para τ_P no intervalo $[0.005, 0.1]$ e $P_Z = 0.25$.

τ_P	P_Z				
	0	0.25	0.5	0.75	1
0.05	1.31779	0.00322794	0.00312518	0.00319808	0.00309818
0.1	1.44865	0.00311461	0.00303908	0.00308612	0.0034524
0.2	1.26156	0.00309722	0.0030462	0.0031066	0.0032642
0.3	1.339	0.00311028	0.00307859	0.00313666	0.00373456
0.5	1.1804	0.00319829	0.00316559	0.00324735	0.00391438

Tabela 5.12: Valores médios da FO (erro RMS) obtidos para diversas combinações dos parâmetros τ_P e P_Z para o projeto de filtro refletor simples.

τ_P	P_Z				
	0	0.25	0.5	0.75	1
0.05	2.04165	0.0431816	0.835129	0.931543	1.37906
0.1	2.03631	0.616303	1.19266	0.965796	1.46834
0.2	2.08032	1.25691	1.32229	1.48647	1.3537
0.3	2.04029	0.943126	1.28174	1.01181	1.41328
0.5	2.06999	1.0999	1.42079	1.24018	1.30823

Tabela 5.13: Valores médios da FO (erro RMS) obtidos para diversas combinações dos parâmetros τ_P e P_Z para o projeto de filtro rampa.

Apenas 5 amostras foram empregadas no cálculo das médias apresentadas nas Tabelas 5.12, 5.13 e nas figuras 5.24 e 5.25. Apesar deste número reduzido de amostras alargar indesejavelmente os intervalos de confiança, ainda assim é possível reconhecer certas tendências nos resultados. Primeiramente, a qualidade dos filtros parece sempre reduzir quando o valor de P_Z se aproxima de zero ou de 1. Em segundo lugar, a qualidade dos filtros parece sempre reduzir para valores de τ_P próximos de 0.5 para ambos os projetos. Particularmente, para o projeto rampa, o limite inferior para τ_P não foi identificado embora seja óbvio que para $\tau_P = 0$ o EPSO deva se comportar (indesejavelmente) como um PSO simples (não auto-adaptativo). Maiores diferenças entre os resultados obtidos para os dois projetos podem ser percebidas ao se comparar as figuras 5.24 e 5.25. O projeto de refletor simples parece oferecer melhores resultados para P_Z entre 0.4 e 0.7 e para τ_P entre 0.1 e 0.2; enquanto o projeto rampa oferece melhores resultados para P_Z entre 0.1 e 0.4 e para τ_P entre a 0.005 e 0.1. O projeto rampa

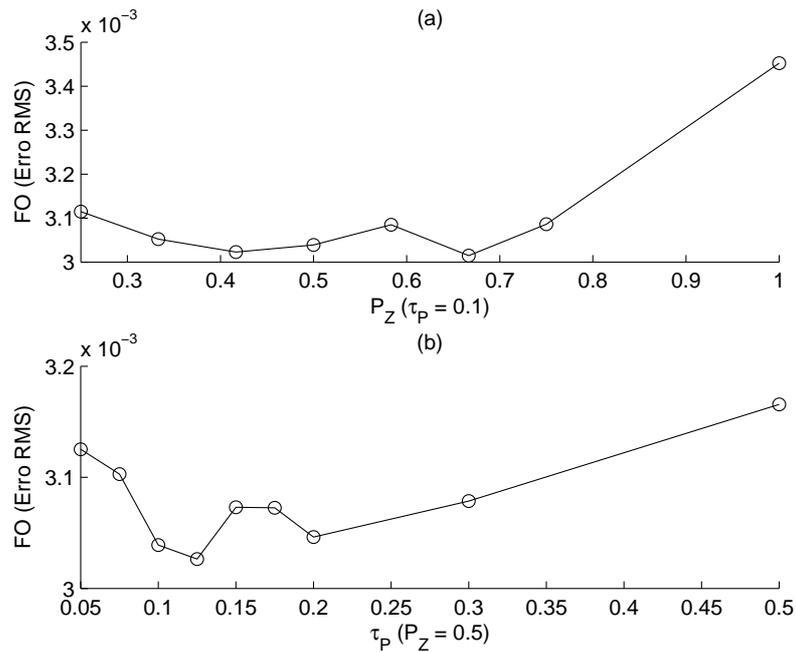


Figura 5.24: Curvas de erro RMS obtidas para o projeto de filtro refletor simples: (a) em função de P_Z para $\tau_P = 0.1$ e (b) em função de τ_P para $P_Z = 0.5$ (b).

claramente necessita de menores probabilidades de cooperação e desvio de mutação, sendo consideravelmente mais sensível à variações destes parâmetros do que o projeto do filtro refletor simples. Incluir P_Z no conjunto de variáveis auto-adaptadas poderia constituir uma opção bastante razoável.

5.2.2 Técnicas de decodificação

As técnicas de decodificação introduzidas para o GA nas seções anteriores promovem alterações no modo como a heurística interpreta o espaço de busca. Esta alteração, de um modo geral, promove uma simplificação do espaço de busca, permitindo a localização de soluções promissoras por parte da heurística com menos esforço. O efeito das técnicas de decodificação foi satisfatoriamente demonstrado para o GA nas seções anteriores. Porém, estas técnicas não estão exclusivamente associadas a quaisquer particularidades do GA, sendo portanto razoável esperar resultados similares caso aplicadas ao PSO.

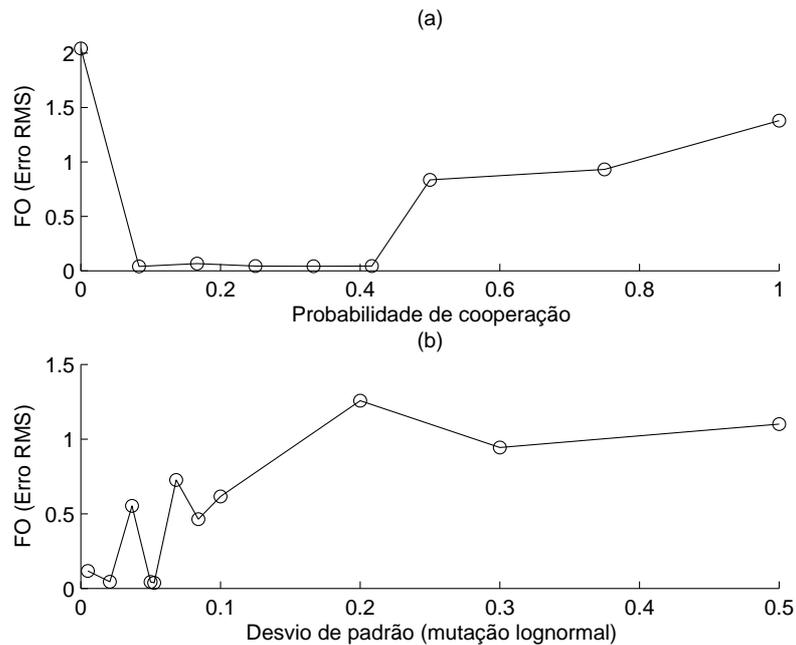


Figura 5.25: Curvas de erro RMS obtidas para o projeto de filtro rampa: (a) em função de P_Z para $\tau_P = 0.05$ e (b) em função de τ_P para $P_Z = 0.25$ (b).

5.2.2.1 Decodificação suavizada

Para demonstrar a aplicação da decodificação suavizada, foram empregados os mesmos projetos de filtro refletor simples e filtro rampa explorados nas seções anteriores. Os parâmetros do EPSO: tamanho do enxame igual a 30 e 5000 iterações. Particularmente, para o projeto de filtro refletor simples, foram empregados $\tau_P = 0.1$ e $P_Z = 0.5$, enquanto que, para o projeto de filtro rampa, foram empregados $\tau_P = 0.05$ e $P_Z = 0.25$ (valores estimados a partir dos resultados coletados na seção anterior). Os parâmetros de decodificação foram os mesmos empregados para o GA nas seções anteriores: $p = \{0.02, 0.05, 0.1\}$ para a DSR e $span = \{2, 5, 10\}$ para a DSMM. Os resultados são mostrados nas tabelas 5.14 e 5.15 respectivamente para o projeto refletor simples e rampa. Para o projeto refletor simples, a Tabela 5.14 indica que o uso de técnicas de decodificação suavizadas não implicam em resultados melhores do que o EPSO empregando decodificação convencional. Entretanto, a Fig. 5.26 mostra que os resultados típicos obtidos com a DSR com $p = 0.1$ e com a decodificação convencional são indistinguíveis. Para o projeto rampa, entretanto, ocorre algo inverso. O EPSO com decodificação suavizada consegue melhores resultados do que o EPSO com decodificação convencional, o que é mostrado na Fig. 5.27. De modo geral, a decodificação suavizada oferece resultados satisfatórios combinados a uma maior confiabilidade do EPSO, o que pode ser comprovado através dos valores de desvio padrão obtidos para as tabelas 5.14 e

5.15.

Decodificação	FO			Convergência
	Desvio	Média	Melhor	média
$p = 0.02$	0.0003702	0.2795172	0.2791203	2135
$p = 0.05$	0.0001541	0.0210927	0.0209339	3928
$p = 0.1$	0.0001535	0.0036364	0.0033569	4542
$span = 2$	0.0004138	0.2793684	0.2790768	2643
$span = 5$	0.0000791	0.0210348	0.0209513	4044
$span = 10$	0.0008227	0.0037969	0.0033892	4263
Não suavizado	0.0001005	0.0030850	0.0029944	4500

Tabela 5.14: Comparação entre o EPSO utilizando codificação *smooth* para três diferentes valores de $span$, o EPSO utilizando codificação restritiva para três valores de p e o EPSO não suavizado para o projeto de refletor simples.

5.2.2.2 Decodificação *spline*

Para demonstrar a viabilidade das técnicas de decodificação aproximada empregando *splines* aplicadas combinadas ao EPSO, os projetos de filtro refletor simples e filtro rampa foram utilizados novamente. A técnica DSQ foi aplicada com 5 e 7 pontos de controle. As tabelas 5.16 e 5.17 resumem os resultados obtidos respectivamente para o projeto refletor simples e filtro rampa. Os termos “DSQ5” e “DSQ7” representam respectivamente a DSQ utilizando 5 e 7 pontos mestres. Embora os valores obtidos pela FO sejam consideravelmente maiores do que os observados nas tabelas 5.14 e 5.15, o número de iterações médio necessário para se obter a convergência são drasticamente menores, até 10 vezes para 5 pontos mestres. Além disso, as figuras 5.28 e 5.29 mostram resultados bastante satisfatórios, bem similares aos encontrados utilizando DSR e utilizando a DSMM.

5.2.3 Aceleração do EPSO por meio da redução da quantidade de amostras da FO

Assim como para o GA, no Capítulo 4 também foi proposto um algoritmo EPSO acelerado (Fig. 4.13). O objetivo desta seção é investigar tal algoritmo.

Decodificação	FO			Convergência
	Desvio	Média	Melhor	média
$p = 0.02$	0.0037124	0.1538858	0.1487784	3496
$p = 0.05$	0.0058086	0.0579202	0.0457512	4583
$p = 0.1$	0.0075904	0.0392429	0.0301083	4757
$span = 2$	0.0051007	0.1575607	0.1509556	4071
$span = 5$	0.0126897	0.0599193	0.0440668	4345
$span = 10$	0.0121061	0.0452939	0.0259345	4726
Não suavizado	0.6085971	0.7248780	0.0227847	4707

Tabela 5.15: Comparação entre o EPSO utilizando codificação *smooth* para três diferentes valores de $span$, o EPSO utilizando codificação restritiva para três valores de p e o EPSO não suavizado para o projeto rampa.

A Fig. 5.30 compara curvas de erro RMS entre o espectro de refletividade obtido e o desejável para os filtros ópticos sintetizados pelo EPSO em função do número de amostras, utilizando três diferentes estratégias de amostragem. As curvas de erro exibidas em Fig. 5.30 (a) foram obtidas para o projeto de filtro refletor simples enquanto (b) exibe curvas obtidas para o projeto rampa. O rótulo “A” identifica as curvas obtidas empregando amostragem convencional no cálculo da FO (amostras fixas e uniformemente espaçadas). O rótulo “B” identifica as curvas obtidas empregando amostragem AAL. O rótulo “C” identifica curvas obtidas pelo EPSO acelerado, empregando valores de S_R dado pelo eixo das abscissas e $S_S = 100$. É interessante observar que os valores de erro RMS se estabilizam a medida que o número de amostras tende a 100. Porém o EPSO acelerado parece exibir sempre os mesmos valores de erro RMS, valores estes sempre próximos aos menores obtidos pelas curvas “A” e “B”.

A Fig. 5.31 exibe curvas de refletividade (a) e de perfil de ν (b) para uma FBG típica obtida pelo EPSO acelerado utilizando $S_R = 1$ e $S_S = 100$ para o projeto refletor simples. Estas curvas são praticamente idênticas às obtidas pelo EPSO não acelerado utilizando 100 amostras fixas. Para o projeto rampa, a Fig. 5.32 compara curvas típicas de refletividade (a), de perfil de δn_{eff} (b) e de perfil de λ_b (c) obtidas pelo EPSO não acelerado com 100 amostras fixas (rótulo “A”) e pelo EPSO acelerado (rótulo “B”) utilizando $S_R = 1$ e $S_S = 100$. Ambas as figuras demonstram que o EPSO acelerado utilizando apenas 1 amostra móvel é capaz de alcançar resultados bastante satisfatórios.

A Fig. 5.33 analisa os ganhos de aceleração obtidos pelo EPSO modificado. Em 5.33 (a) os ganhos foram calculados com base no tempo médio de processamento de apenas uma

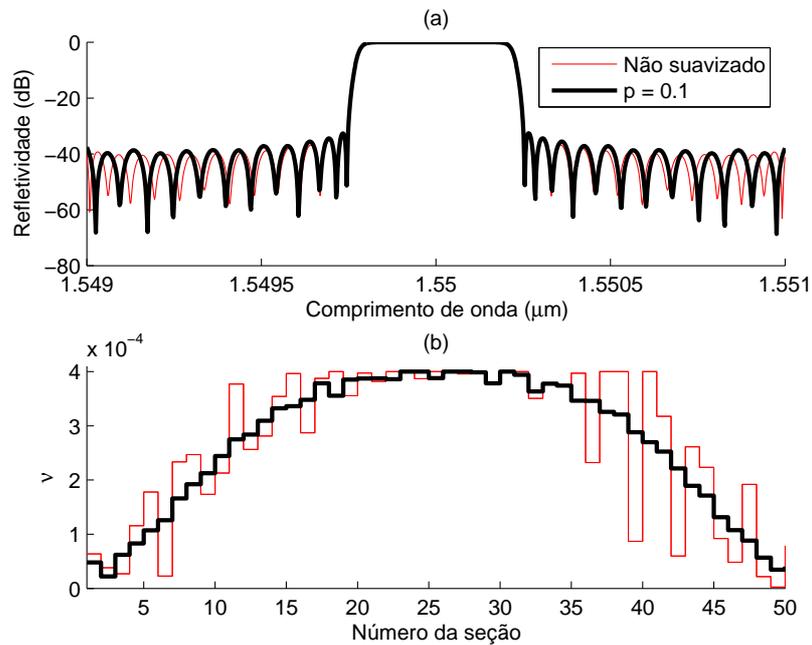


Figura 5.26: Curvas de refletividade em função do comprimento de onda (a) e de perfil de visibilidade de franjas ν em função do número da seção (b) para duas grades: uma obtida pelo EPSO sem decodificação suavizada e outra com DSR com $p = 0.1$.

iteração. Os resultados são bem próximos ao ideal previsto por (4.27) para $S_{Ri} = S_R$ e $N_\lambda = 2$. Em 5.33 (b) os ganhos foram calculados considerando o tempo de processamento total. Tal como observado para o GA acelerado, o EPSO acelerado também obtém ganhos práticos expressivos, porém bastante inferiores aos ideais. Particularmente para o projeto do filtro refletor, os ganhos foram muito menores do que os observados para o GA, sendo para este caso recomendável o uso de $S_R = 1$.

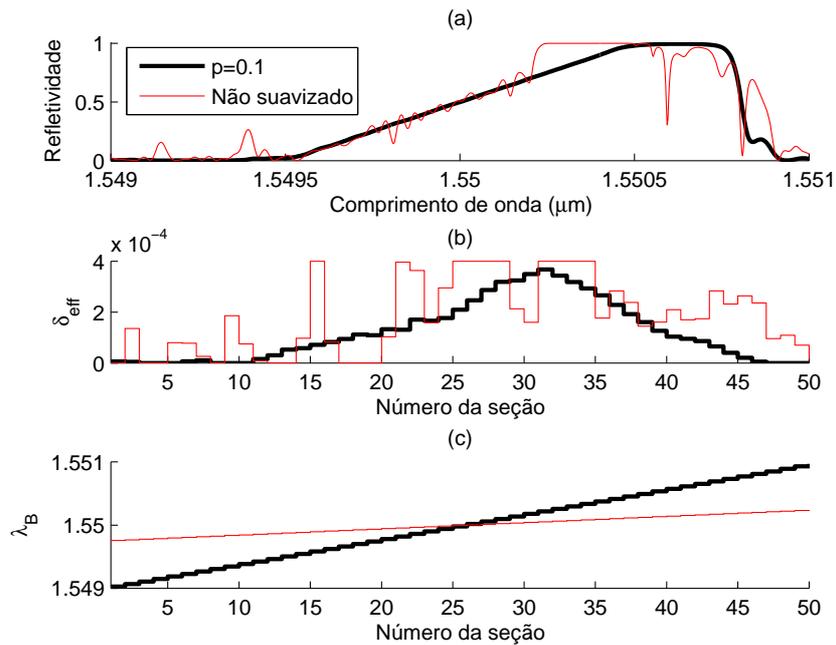


Figura 5.27: Curvas de refletividade em função do comprimento de onda (a) e de perfil de visibilidade de franjas ν em função do número da seção (b) para duas grades: uma obtida pelo EPSO sem decodificação suavizada e outra com DSR com $p = 0.1$.

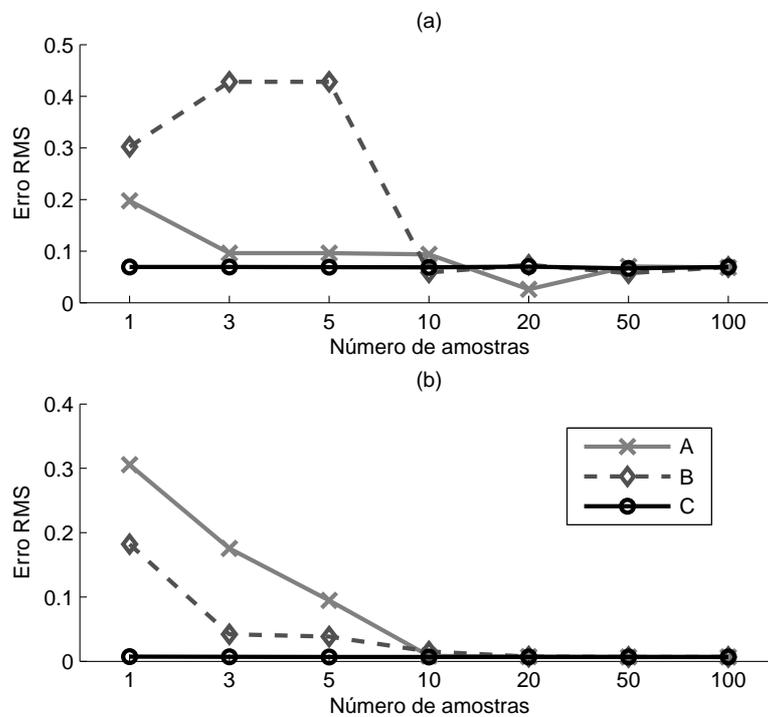


Figura 5.30: Curvas de erro RMS em função do número de amostras para três tipos de amostragem, para o projeto de filtro refletor (a) e projeto de filtro rampa (b).

População	Decod.	FO			Convergência
		Desvio	Média	Melhor	média
5	DSQ5	0	0.0179626	0.0179626	65
10	DSQ5	0	0.0179626	0.0179626	58
20	DSQ5	0	0.0179626	0.0179626	37
30	DSQ5	0	0.0179626	0.0179626	28
5	DSQ7	0	0.0226543	0.0226543	218
10	DSQ7	0	0.0226543	0.0226543	162
20	DSQ7	0	0.0226543	0.0226543	89
30	DSQ7	0	0.0226543	0.0226543	88

Tabela 5.16: Resultados obtidos empregando DSQ com 5 e 7 pontos mestres sobre o projeto refletor simples.

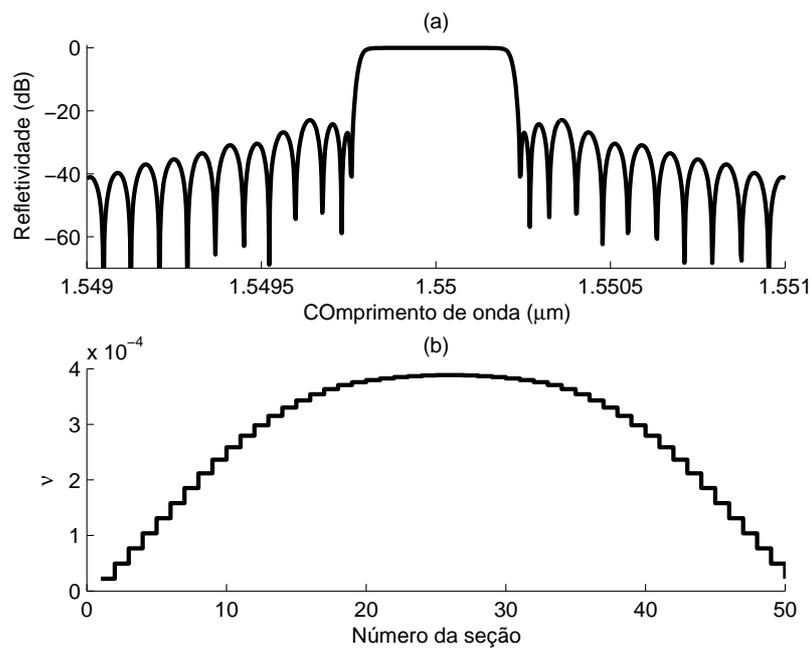


Figura 5.31: Curva de refletividade para um filtro refletor um obtido pelo EPSO acelerado (a) e seu respectivo perfil de ν (b).

População	Decod.	FO			Convergência
		Desvio	Média	Melhor	média
5	QSP5	0	0.0724985	0.0724985	1189
10	QSP5	0	0.0724985	0.0724985	847
20	QSP5	0	0.0724985	0.0724985	605
30	QSP5	0	0.0724985	0.0724985	562
5	QSP7	0.0095515	0.1010761	0.0768146	1338
10	QSP7	0.0137640	0.0967467	0.0768007	1562
20	QSP7	0.0132936	0.0871726	0.0768007	3675
30	QSP7	0.0120726	0.0854569	0.0768007	2529

Tabela 5.17: Resultados obtidos empregando DSQ com 5 e 7 pontos mestres sobre o projeto rampa.

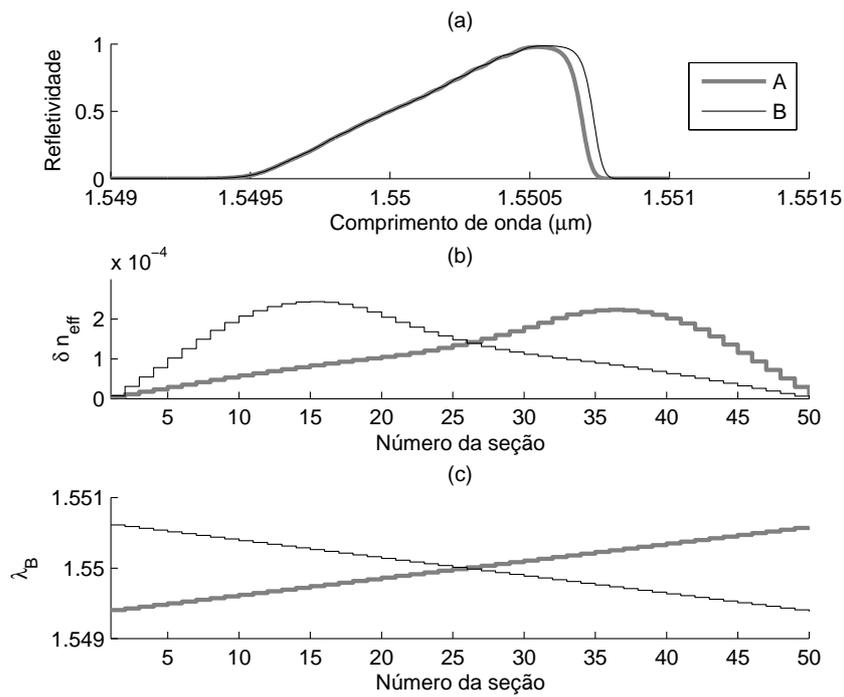


Figura 5.32: Curvas de refletividade para dois filtros rampa, um obtido pelo EPSO convencional e outro pelo EPSO acelerado (a) e os respectivos perfis de δn_{eff} (b) e de λ_B (c).

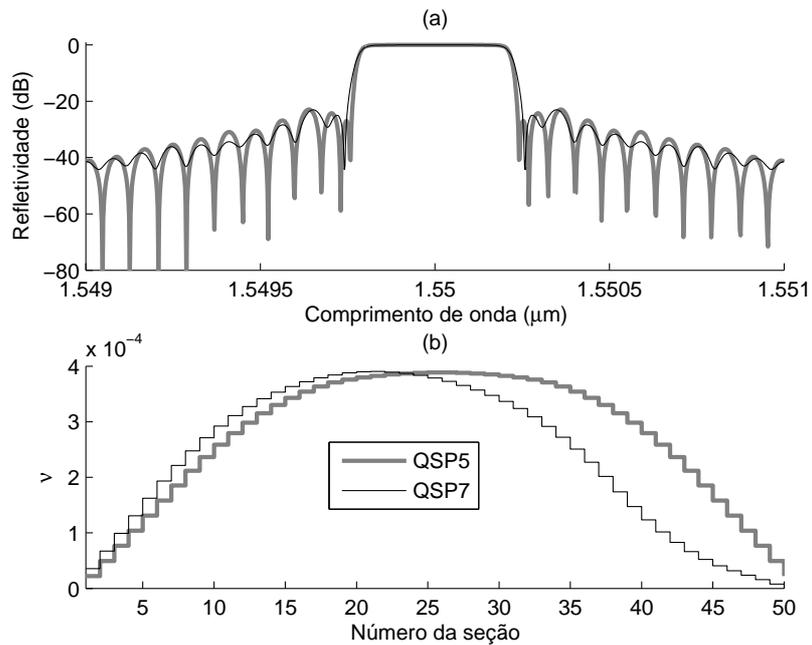


Figura 5.28: Curvas de refletividade em função do comprimento de onda (a) e de perfil de visibilidade de franjas ν para duas grades obtidas pelo EPSO empregando DSQ com 5 e 7 pontos mestres.

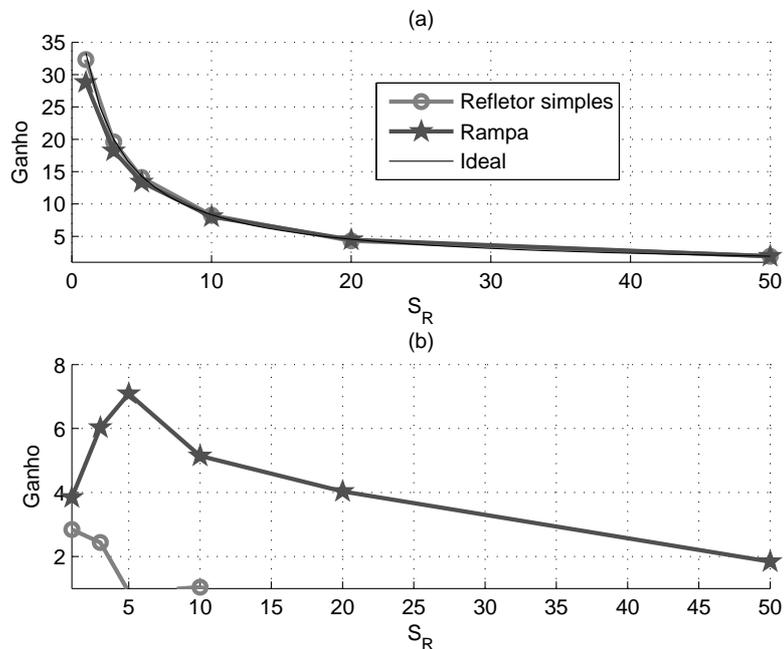


Figura 5.33: Curvas de ganho do EPSO acelerado em relação ao EPSO convencional: curvas obtidas em função do tempo de processamento médio de uma iteração (a); curvas obtidas em função do tempo de processamento total.

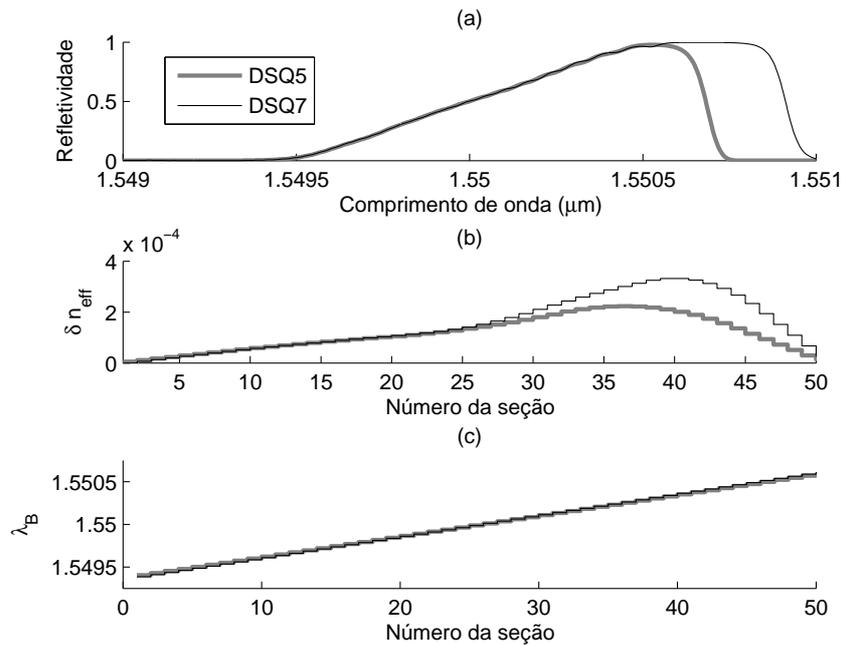


Figura 5.29: Curvas de refletividade em função do comprimento de onda (a), de perfil de perturbação δn_{eff} em função do número da seção (b) e perfil de λ_B para duas grades obtidas pelo EPSO empregando DSQ com 5 e 7 pontos mestres.

5.3 Comparação entre o GA e o EPSO

A maneira mais segura de se comparar as duas heurísticas aplicadas à síntese de filtros ópticos é através do número médio de chamadas à FO. Quanto mais eficiente uma heurística, menor deverá ser o número de chamadas para se alcançar o mesmo resultado. Para o projeto rampa, empregando decodificação *spline* quadrática com 5 pontos de controle, o GA não acelerado, com uma população de 20 indivíduos e empregando FO com 100 amostras fixas, converge em média em aproximadamente 572 gerações (Tabela 5.10). O EPSO equivalente converge em aproximadamente 847 iterações utilizando um enxame de 10 partículas (Tabela 5.17). Para o GA, o número de chamadas à FO por geração é exatamente igual ao tamanho da população. Porém, o EPSO com $N_\lambda = 2$ realiza 3 vezes chamadas por iteração do que o tamanho do enxame, totalizando 30 chamadas por iteração. Portanto, o GA acaba realizando em média 11440 chamadas à FO, enquanto o EPSO realiza em média 25410 chamadas. Portanto, embora ambas as heurísticas apresentem resultados finais similares, o EPSO é menos vantajoso por exigir quase 100% mais chamadas à FO, sendo, portanto, em média 2 vezes mais lento do que o GA. Entretanto, é preciso lembrar que o EPSO é consideravelmente mais simples do que o GA e obteve resultados de qualidade superior. Além disso, o EPSO não

foi tão extensamente investigado quanto foi o GA, principalmente em se tratando de outras combinações de parâmetros. Por exemplo, o EPSO não foi testado para $N_\lambda \neq 2$. Portanto, comparações entre as duas heurísticas parecem ser inconclusivas.

Capítulo 6

Conclusões

Esta última década foi caracterizada pela aplicação extensiva de redes de sensores ópticos, aplicadas desde na construção civil até na indústria aeronáutica. Nestas redes as FBGs ocupam um lugar de destaque, tanto como elementos sensores (ou transdutores) quanto filtros dos sistemas de interrogação. Quanto maior os desafios impostos às redes ópticas de sensoriamento, maior a sofisticação das FBGs ou maior a criatividade com que devem ser combinadas na forma de sistemas robustos e eficientes. Particularmente as FBGs empregadas na interrogação têm sido modificadas para permitir maior precisão de aferições dentro da maior faixa possível da grandeza de interesse. As FBGs com perfis de refletividade triangulares são as que mais caracterizam esta tendência.

Existem, entretanto, sistemas ópticos de sensoriamento que consistem exclusivamente em filtros ópticos, onde grades transdutoras simplesmente não existem. Estes são tipicamente sistemas de detecção ou medição de compostos ou substâncias químicas, onde as assinaturas espectrais são empregadas para qualificar e quantificar. As substâncias podem refletir, transmitir, espalhar, absorver e emitir (por exemplo, através da fluorescência) mais ou menos luz em função do tempo e da frequência. Filtros especiais baseados em FBGs podem refletir ou transmitir trechos característicos do espectro seguindo um padrão específico de acordo com a substância de interesse. Esta espécie de sensor óptico pode necessitar apresentar espectros de refletividade ou transmissividade bastante arbitrários, cujo projeto pode ser desafiador.

A combinação de exigências de projetos de filtros FBG mais sofisticados com as limitações técnicas e econômicas de produção resulta em um campo bastante fértil para as heurísticas, como o GA e o PSO. No esforço de contribuir com algoritmos mais eficientes e de maior desempenho, esta Tese apresentou algumas contribuições:

- **Algoritmos flexíveis** no sentido de poderem seguir diferentes abordagens mais

freqüentes na literatura. Para o GA, é possível que este realize um esquema evolucionário convencional ou possa seguir um esquema baseado em estratégias evolutivas, onde o número de ancestrais e de descendentes possam ser diferentes dentro de uma dada geração. Para o PSO, existe a possibilidade de empregar uma abordagem convencional ou auto-adaptativa, empregando uma topologia estrela ou distribuída. Porém a contribuição mais notável tanto para o GA quanto para o PSO propostos é a capacidade destes de empregar estrategicamente diferentes amostragens para a FO, permitindo assim o uso mais freqüente de amostragens esparsas e assim o aumento substancial do desempenho computacional das heurísticas sem grandes impactos negativos sobre a qualidade das soluções geradas. Essa abordagem abre a possibilidade do uso simultâneo de FOs essencialmente diferentes. Por exemplo, durante a síntese de uma estrutura dielétrica bidimensional é possível empregar um método baseado em diferenças finitas com uma discretização (na medida do possível) relaxada para efetuar uma avaliação mais rápida, constituindo assim o equivalente da FO com amostragem esparsa. Um método baseado em elementos finitos com uma discretização mais fina poderia constituir o equivalente a uma FO com amostragem mais rigorosa. Estruturas mais complexas do que FBGs unidimensionais, como cristais fotônicos, poderiam ser sintetizadas com o uso de recursos computacionais mais modestos.

- **Técnicas de amostragem** AP e AA que permitem o uso de uma quantidade racional de amostras dependendo da complexidade das curvas de refletividade. Particularmente a AA tem o potencial de permitir a otimização de filtros com bandas extremamente estreitas, tais como filtros de fenda feitos com deslocamentos discretos de fase. De um modo geral, a AP e a AA podem permitir uma avaliação rigorosa de FBGs com uma grande economia de amostras e, conseqüentemente, com significativos ganhos de desempenho. Outras técnicas de amostragem apresentadas foram a AD e a AAL, sendo esta última intensivamente utilizada em combinação com o GA e com o PSO acelerados obtendo excelentes resultados.
- **Técnicas de decodificação** especializadas. As decodificações suavizadas permitem limitar significativamente o espaço de soluções do problema, enquanto que as técnicas de decodificação baseadas em interpolação ou em aproximações permitem uma radical redução da dimensionalidade do problema. Esta classe de contribuição permite um aumento do desempenho das heurísticas, muito embora de forma indireta, através da redução da quantidade de populações e iterações necessárias.

Para o GA flexível, a técnica de aceleração baseada no uso de uma FO e duas amostragens permitiu ganhos de *speed up* reais da ordem de 5 vezes. Entretanto o ganho obtido com

o uso de decodificações especializadas foi bem maior: algo em torno de 50 vezes. A viabilização das técnicas de decodificação deve-se em grande parte pela oportunidade do estágio de doutorado (Programa de Doutorado no País com Estágio no Exterior - PDEE, da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES) no Instituto de Engenharia de Sistemas e Computadores do Porto (INESC-Porto - Portugal). Graças ao intercâmbio foi possível conhecer as restrições dos métodos de fabricação e assim alimentar as heurísticas com dados mais realísticos.

Infelizmente nem todas idéias desenvolvidas na Tese foram satisfatoriamente testadas. Por exemplo, as técnicas de amostragem, particularmente a AA, possuem um grande potencial para a redução do custo computacional das heurísticas. Também alguns parâmetros importantes do PSO não foram testados, como o efeito da topologia, se estrela ou distribuída. Estas lacunas podem ser preenchidas em algum prosseguimento deste trabalho. Também outras atividades poderiam ser oportunamente exploradas em um eventual trabalho futuro:

- Aplicação do GA ou PSO acelerados na síntese de dispositivos mais complexos, como superfícies seletoras de frequência ou filtros baseados em cristais fotônicos;
- Comparações entre os algoritmos propostos com outras heurísticas alternativas, como o micro GA;
- Aplicação da amostragem adaptativa na síntese de filtros de fenda criados com deslocamentos discretos de fase;
- Adaptação das heurísticas apresentadas à problemas multiobjetivo.

6.1 Publicações

6.1.1 Artigos publicados em periódicos

- Souza, M. J. , Carvalho, J. C. C. , Sales Jr., C. S. , Segatto, M. V., Costa, J. C. W. A. , “A new acceleration technique for the design of fibre gratings”. *Optics Express*, V. 14, p. 10715 - 10725, 2006.
- Carvalho, J. C. C. , Sousa, M. J. , Sales Jr., C. S. , Costa, J. C. W. A. , Francês, C. R. L. , “Otimização de Redes de Bragg em Fibra Usando Algoritmos Genéticos”. *Journal of Microwaves and Optoelectronics*, V. 4, p. 136 - 151, 2005.

6.1.2 Artigos publicados em conferências

- Souza, M. J. , Sales JR, C. S. , COSTA, J. C. W. A. , “Estratégias de Codificação e Aceleração para Algoritmo Genético aplicado na Síntese de Grades de Bragg em Fibra”. 12º *SBMO - Simpósio Brasileiro de Microondas e Optoeletrônica e 7º CBMAG - Congresso Brasileiro de Eletromagnetismo (MOMAG 2006)*, 2006.
- Carvalho, J. C. C. , Sousa, M. J. , Sales Jr., C S. , COSTA, J. C. W. A. , Francês, C. R. L. , “Optimization of Bragg Grating in Optical Fiber Using Modified Fitness Function and an Accelerated Genetic Algorithm”, *Telecommunications Symposium 2006 (ITS 2006)*, 2006.
- Borges, G. S. , Sousa, M. J. , COSTA, J. C. W. A. , “Técnica de aceleração para AG aplicado à síntese de dispositivos ópticos multicamadas”, *XXVII CILAMCE*, 2006.
- Macêdo, J. A. S, Sousa, M. J. , Dmitriev, Victor , “Optimization of Wide-Band Multilayer Microwave Absorbers for any Angle of Incidence and Arbitrary Polarization”, *Microwave and Optoelectronics Congress*, 2005.
- Sousa, M. J. , Sales Jr., C. S. , Francês, C. R. L., Costa, J. C. W. A. , “Síntese de Filtros Ópticos Utilizando Algoritmo Genético Modificado”, *XXI Simpósio Brasileiro de Telecomunicações*, 2004.

Referências Bibliográficas

- [1] Rui Huang, Yingwu Zhou, Haiwen Cai, Ronghui Qu, and Zujie Fang. A fiber bragg grating with triangular spectrum as wavelength readout in sensor systems. *Optics Communications*, (229):197–201, 2004.
- [2] K. A. Winick and J. E. Roman. Design of corrugated waveguide filters by fourier transform techniques. *IEEE Journal of Quantum Electronics*, 26:1918–1929, 1990.
- [3] Johannes Skaar and Ole Henrik Waagaard. Design and characterization of finite-length fiber gratings. *IEEE Journal of Quantum Electronics*, 39:1238–1245, 2003.
- [4] K. Aksnes and J. Skaar. Design of short fiber bragg gratings by optimization. *Applied Optics*, 43:2226–2230, 2004.
- [5] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, January 1989.
- [6] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proceedings of IEEE Conference of Neural Networks IV*, 1989.
- [7] Johannes Skaar and Knut Magne Risvik. A genetic algorithm for the inverse problem in synthesis of fiber gratings. *IEEE Journal of Lightwave Technology*, 16:1928–1932, 1998.
- [8] S. Baskar, A. Alphones, P. N. Suganthan, N. Q. Ngo, and R. T. Zheng. Design of optimal length low-dispersion FBG filter using covariance matrix adapted evolution. *IEEE Photonics Technology Letters*, 17:2119–2121, 2005.
- [9] S. Baskar, R. T. Zheng, A. Alphones, N. Q. Ngo, and P. N. Suganthan. Particle swarm optimization for the design of low-dispersion fiber bragg gratings. *IEEE Photonics Technology Letters*, 17:615–617, 2005.
- [10] S. Baskar, P. N. Suganthan, N. Q. Ngo, A. Alphones, and R. T. Zheng. Design of triangular FBG filter for sensor applications using covariance matrix adapted evolution algorithm. *Optics Communications*, (260):716–722, 2006.

-
- [11] Charles M. Jewart. *A Highly-Sensitive Fiber Bragg Grating Transverse Strain Sensor Using Micro-Strucute Fibers*. PhD thesis, Pittsburgh University, 2006.
- [12] K. O. Hill, Y. Fujii, D. C. Johnson, and B. S. Kawasaki. Photosensitivity in optical fiber waveguides: application to reflection fiber fabrication. *Applied Physics Letters*, 32:647–649, 1978.
- [13] G. Meltz, W. W. Morey, and W. H. Glenn. Formation of bragg gratings in optical fibers by a transverse holographic method. *Applied Physics Letters*, 14:823–825, 1989.
- [14] B. Malo, K. O. Hill, F. Bilodeau, D. C. Johnson, and J. Albert. Point-by-point fabrication of micro-bragg gratings in photosensitive fiber using single excimer pulse refractive-index modification techniques. *Electronic Letters*, 29:1668–1669, 1993.
- [15] J. Carlos C. da Silva. *Monitoração de Vibrações de Estruturas com o Emprego de Sensores em Fibra Óptica*. PhD thesis, CEFET-PR, 2005.
- [16] Turan Erdogan. Fiber grating spectra. *Journal of Lightwave Technology*, 15:1277–1294, 1997.
- [17] Dietrich Marcuse. *Theory of Dielectric Optical Waveguides*. Academic press, 1974.
- [18] M. Yamada and K. Sakuda. Analysis of almost-periodic distributed feedback slab waveguides via a fundamental matrix approach. *Applied Optics*, 26:3474–3478, 1987.
- [19] Spectral grids for WDM applications: DWDM frequency grid, recommendation ITU-T G.694.1, 2002.
- [20] Govind P. Agrawal. Phase-shifted fiber bragg gratings and their application for wavelength demultiplexing. *IEEE Photonics Technology Letters*, 6(8):995–997, 1994.
- [21] Christophe Martinez and Pierre Ferdinand. Analysis of phase-shifted fiber bragg gratings written with phase plates. *Applied Optics*, 38(15):3223–3228, 1999.
- [22] M. Ragulskis and L. Saunoriene. Applicability of optical geometric differentiation for time-average geometric moire. *Blackwell Publishing Ltd*, 42:173–179, 2006.
- [23] Andreas Othonos and Xavier Lee. Novel and improved methods of writing bragg gratings with phase masks. *IEEE Photonics Technology Letters*, 7(10):1183–1185, 1995.
- [24] E. Peral, J. Capmany, and J. Marti. Iterative solution to the gelfand-levitan-marchenko equations and application to synthesis of fiber gratings. *IEEE Journal of Quantum Electronics*, 32:2078–2084, 1996.

-
- [25] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 1992.
- [26] Edward E. Altshuler and Derek S. Linden. An ultrawide-band impedance-loaded genetic antenna. *IEEE Transactions On Antennas And Propagation*, 52:3147–3149, 2004.
- [27] MathWorks. Rastrigin’s function: Getting starter with the genetic algorithm, 12 2007.
- [28] David E. Goldberg. Real-coded genetic algorithms, virtual alphabets, and blocking. Technical Report 61801, Department of General Engineering University of Illinois at Urbana-Champaign Urbana, 2001.
- [29] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons LTD, 2001.
- [30] L. Darrell Whitley, Monte Lunacek, and Artem Sokolov. Comparing the niches of CMA-ES, CHC and pattern search using diverse benchmarks. *Lecture Notes in Computer Science*, 4193:988–997, 2006.
- [31] Gregorio Toscano Pulido and Carlos A. Coello Coello. The micro genetic algorithm 2: Towards on-line adaptation in evolutionary multiobjective optimization. In *Evolutionary Multi-Criterion Optimization - Second International Conference*, volume 2632, pages 252–266, 2003.
- [32] Sourav Chakravarty, Raj Mittra, and Neil RhodesWilliams. On the application of the microgenetic algorithm to the design of broad-band microwave absorbers comprising frequency-selective surfaces embedded in multilayered dielectric media. *IEEE Transactions on Microwave Theory and Techniques*, 49(6), 2001.
- [33] Gabriel Alvarez. Can we make genetic algorithms work in high-dimensionality problems? Technical Report 112, Stanford University Exploration Project, 2002.
- [34] Tian Li Yu, Kumara Sastry, and David E. Goldberg. On-line population size adjusting using noise and substructural measurements. Technical Report 2005017, Department of General Engineering University of Illinois at Urbana-Champaign Urbana, 2005.
- [35] Thomas Bäck and Schwefel. Evolutionary computation: an overview. In *International Conference on Evolutionary Computation*, 1996.
- [36] William M. Spears. Adapting crossover in evolutionary algorithms. In *Evolutionary Programming*, pages 367–384, 1995.

-
- [37] Manuel Lozano, Francisco Herrera, Natalio Krasnogor, and Daniel Molina. Real coded memetic algorithms with crossover hill-climbing. *Evolutionary Computation*, 12:273–302, 2004.
- [38] Zhiming Liu, Jiliu Zhou, and Su Lai. New adaptive genetic algorithm based on ranking. In *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, pages 1841–1844, 2003.
- [39] Thomas B. and H-P. Schwefel. Evolutionary computation: comments on the history and current state. *IEEE Transactions on Evolutionary Computation*, 1:3–17, 1997.
- [40] M. Affenzeller and S. Wagner. Offspring selection: A new self-adaptive selection scheme for genetic algorithms. *Adaptive and Natural Computing Algorithms*, pages 218–221, 2005.
- [41] Marcus Hutter. Fitness uniform selection to preserve genetic diversity. In *Proceedings of the Congress on Evolutionary Computation, CEC'02*, pages 783–788, 2002.
- [42] A. E. Eiben, M. C. Schut, and A. R. de Wilde. Boosting genetic algorithms with self-adaptive selection. *IEEE Congress on Evolutionary Computation, CEC 2006.*, pages 477–482, 2006.
- [43] Jacob Robinson and Yahya Rahmat-Samii. Particle swarm optimization in electromagnetics. *IEEE Transactions on Antennas and Propagation*, 52(2):397–407, 2004.
- [44] Mihaela Breaban and Henri Luchian. PSO under an adaptive scheme. *The 2005 IEEE Congress on Evolutionary Computation*, 2:1212–1217, 2005.
- [45] Xiaohui Hu and R. C. Eberhart. Adaptive particle swarm optimization: detection and response to dynamic systems. In *Proceedings of the 2002 Congress on Evolutionary Computation CEC '02*, volume 2, pages 1666–1670, 2002.
- [46] D. J. Krusienskia and W. K. Jenkins. A modified particle swarm optimization algorithm for adaptive filtering. In *IEEE International Symposium on Circuits and Systems, ISCAS 2006*, 2006.
- [47] Vladimiro Miranda and Nuno Fonseca. EPSO – best-of-two-worlds meta-heuristic applied to power system problems. In *Evolutionary Computation 2002 CEC'02 Proceedings of the 2002 Congress*, volume 2, pages 1080–1085, 2002.
- [48] Maurice Clerc. TRIBES, a parameter free particle swarm optimizer. In *Optimisation par Essaim Particulaire OEP'03*, 2003.

- [49] Vladimiro Miranda and Naing Win Oo. New experiments with EPSO – evolutionary particle swarm optimization. In *Proceedings of the IEEE Swarm Optimization Symposium*, 2005.
- [50] Rui Mendes. *Population topologies and their influence in particle swarm performance*. PhD thesis, Departamento de informática, Escola de Engenharia, Universidade de Minho, 2004.
- [51] Stefan Janson and Martin Middendorf. A hierarchical particle swarm optimizer and its adaptive variant. *IEEE Transactions on Systems, MAN, and Cybernetics*, 35:1272–1282, 2005.
- [52] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes in C++*. Cambridge University Express, 3 edition, 2002.
- [53] E. Herbert Li and A. B. Djuri. Genetic algorithm for reflective filters design. In *the Hong Kong Devices Meeting*, 2001.