



Programa Interdisciplinar de Pós-Graduação em

Computação Aplicada

Mestrado Acadêmico

Maurício Harff

Aplicação de Metaheurísticas no Desenvolvimento de um
Modelo de Otimização para o Processo de Codificação de
Áudio do Sistema Brasileiro de Televisão Digital

São Leopoldo, 2013

MAURÍCIO HARFF

**Aplicação de Metaheurísticas no Desenvolvimento
de um modelo de Otimização para o Processo de
Codificação de Áudio do Sistema Brasileiro de
Televisão Digital**

Dissertação apresentada como requisito parcial
para a obtenção do título de Mestre pelo Programa
Interdisciplinar de Pós-Graduação em Computação
Aplicada da Universidade do Vale do Rio dos Sinos.

Orientador: Prof. Dr. Arthur Tórgo Gómez

São Leopoldo

2013

Ficha catalográfica

H278a Harff, Maurício

Aplicação de metaheurísticas no desenvolvimento de um modelo de otimização para o processo de codificação de áudio do Sistema Brasileiro de Televisão Digital / por Maurício Harff. – 2013.

193 f. : il., 30cm.

Dissertação (mestrado) — Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, 2013.

Orientação: Prof. Dr. Arthur Tórgo Gómez.

1. Sistema Brasileiro de Televisão Digital - SBTVD.
2. Compressão de áudio. 3. MPEG-4 AAC. 4. Metaheurísticas.
5. Busca tabu. 6. Algoritmo genético. I. Título.

CDU 004.421

Catálogo na Fonte:
Bibliotecária Vanessa Borges Nunes - CRB 10/1556

AGRADECIMENTOS

Ao meu professor orientador Dr. Arthur Tórgo Gómez pela atenção dedicada, colaborando com ideias, sugestões e norteamento para o desenvolvimento do trabalho, além do constante incentivo.

À minha esposa que está sempre ao meu lado em todos os momentos, pela paciência, compreensão e companheirismo.

Ao meu colega de mestrado Roberto Gomes, pelas boas conversas e trocas de ideias a respeito do trabalho.

RESUMO

A qualidade perceptual alcançada pelos codificadores de áudio depende diretamente da escolha de seus parâmetros. O codificador MPEG-4 AAC (*Advanced Audio Coding*), utilizado no Sistema Brasileiro de Televisão Digital (SBTVD), possui em sua estrutura uma etapa composta por um laço de iteração para escolher os parâmetros do codificador, de maneira dinâmica durante o processo de codificação. Este processo de escolha pode ser definido como um problema de Pesquisa Operacional, sendo um problema de Seleção de Partes, denominado como o Problema de Codificação AAC. A estrutura existente no codificador de referência, não resolve este problema de maneira ótima. Desta forma, este trabalho propõe o desenvolvimento e implementação de um modelo de uma estrutura de simulação, para encontrar os parâmetros do codificador de áudio MPEG-4 AAC, de maneira a otimizar a qualidade perceptual do áudio, para uma determinada taxa de bits (*bit rate*). A implementação da estrutura de otimização foi desenvolvida em linguagem C, utilizando as metaheurísticas Busca Tabu e Algoritmo Genético em uma estrutura híbrida. Através da minimização da métrica *ANMR* (*Average Noise-to-Mask Ratio*), o algoritmo procura identificar a melhor configuração dos parâmetros internos do codificador MPEG-4 AAC, de maneira que possa garantir uma qualidade perceptual para o sinal áudio. Os resultados obtidos utilizando a estrutura híbrida de otimização apresentaram valores menores para a métrica *ANMR*, ou seja, uma melhor qualidade perceptual de áudio, quando comparados com os resultados obtidos com o codificador de referência MPEG-4 AAC.

Palavras chave: Sistema Brasileiro de Televisão Digital (SBTVD), compressão de áudio, MPEG-4 AAC (*Advanced Audio Coding*), metaheurísticas, Busca Tabu, Algoritmo Genético.

ABSTRACT

The perceptual quality achieved by audio encoders depends directly on the choice of its parameters. The MPEG-4 AAC (Advanced Audio Coding), used in the Brazilian Digital Television System (BDTS), has a step in its structure that consists in iteration loop to choose the parameters of the encoder dynamically during the encoding process. This selection process can be defined as a problem of Operational Research, being a Part Selection Problem, termed as AAC Encoding Problem. The structure in the reference encoder not solves this problem optimally. Thus, this paper proposes the development and implementation of a model simulation of a structure, to find the internal parameters of the MPEG-4 AAC audio encoder, so as to optimize the perceptual audio quality for a given bit rate. The implementation of the optimization framework was developed in ANSI C programming language, using the Tabu Search and Genetic Algorithm metaheuristics in a hybrid structure. Through the minimization of the ANMR (Average Noise-to-Mask Ratio) metric, the algorithm tries to identify the best configuration of internal parameters of the MPEG-4 AAC. The results obtained using the optimization hybrid structure achieve lower values for the ANMR metric, i.e., an better perceptual audio quality, compared with the obtained with the reference encoder MPEG-4 AAC.

Keywords: Brazilian Digital Television System (SBTVD), audio compression, MPEG-4 AAC (Advanced Audio Coding), metaheuristics, Tabu Search, Genetic Algorithm.

LISTA DE ABREVIATURAS

AAC	<i>Advanced Audio Coding</i>
AG	<i>Algoritmo Genético</i>
ANMR	<i>Average Noise-to-Mask Ratio</i>
ATSC	<i>Advanced Television Systems Committee</i>
AVC	<i>Advanced Video Coding</i>
BT	<i>Busca Tabu</i>
CODEC	<i>Codificação e decodificação de sinais fonte</i>
CD	<i>Compact Disk</i>
DCT	<i>Transformada Discreta do Cosseno</i>
DAB	<i>Difusão de Áudio Digital</i>
ESC	<i>Special Codebook</i>
FLAC	<i>Free Lossless Audio Codec</i>
FO	<i>Função Objetivo</i>
HCb	<i>Huffman Code Book</i>
HE	<i>High Efficiency</i>
ISDB-Tb	<i>Integrated Services Digital Broadcasting Terrestrial</i>
ISO / IEC	<i>International Standards Organization / International Electrotechnical Commission</i>
ITU	<i>International Telecommunication Union</i>
KBD	<i>Kaiser-Bessel Derived</i>
LAV	<i>Largest Absolute Value</i>
LC	<i>Low Complexity</i>
MDCT	<i>Transformada Discreta do Cosseno Modificada</i>
MILP	<i>Mixed Integer Linear Program</i>
MNMR	<i>Maximum Average Noise-to-Mask Ratio</i>
NMR	<i>Noise-to-Mask Ratio</i>
MPEG	<i>Moving Picture Expert Group</i>
PCM	<i>Pulse Code Modulation</i>
PCU	<i>Processor Complexity Units</i>
PS	<i>Parametric Stereo</i>
RAM	<i>Random Access Memory</i>
RCU	<i>RAM Complexity Units</i>

SBR	<i>Spectral Band Replication</i>
SBTVD	Sistema Brasileiro de Televisão Digital
SF	<i>Scale Factor (Fator de Escala)</i>
SFB	<i>Scale Factor Band (Banda do Fator de Escala)</i>
SMR	<i>Signal-to-Mask Ratio</i>
SNR	<i>Signal-to-Noise Ratio</i>
SPL	<i>Sound Pressure Level</i>
TLS	<i>Two Loop Search</i>
TNS	<i>Temporal Noise Shaping</i>
WiFi	<i>Wireless Fidelity</i>
WiMAX	<i>Worldwide Interoperability for Microwave Access</i>
UNISINOS	Universidade do Vale do Rio dos Sinos

LISTA DE FIGURAS

Figura 1 – Sinal contínuo e sinal amostrado	19
Figura 2 – Faixa de frequência do sinal $x_c(t)$	21
Figura 3 – Espectro da função de amostragem.....	21
Figura 4 – Espectro do sinal amostrado com $\Omega_s > 2\Omega_0$	21
Figura 5 – Espectro do sinal amostrado com $\Omega_s < 2\Omega_0$	22
Figura 6 – Sinal Quantizado	24
Figura 7 – Diagrama do Codificador MPEG-4 AAC.....	30
Figura 8 – Limiar absoluto de audição do ouvido humano	33
Figura 9 – Efeito de mascaramento no domínio da frequência	34
Figura 10 – Efeito de mascaramento no domínio do tempo.....	35
Figura 11 – Análise de banda crítica	36
Figura 12 - Divisão de um sinal de áudio dinâmico em frames	40
Figura 13 – Função de janelamento Seno	42
Figura 14 – Sinal amostrado e o mesmo multiplicado pela janela Seno	43
Figura 15 – Estrutura dos blocos no MPEG-4 AAC.....	44
Figura 16 – Exemplo de agrupamento de 8 <i>short blocks</i>	47
Figura 17 – Laço de Iteração AAC	48
Figura 18 – Iteração do laço interno.....	49
Figura 19 – Iteração do laço externo.....	50
Figura 20 – Exemplo de um processo de construção dos Códigos de Huffman	52
Figura 21 – Perfis do codificador de áudio AAC.....	55
Figura 22 – Medida de qualidade objetiva através do <i>NMR</i>	57
Figura 23 – Estratégia utilizada pela Busca Tabu	61

Figura 24 – Pseudocódigo do algoritmo Busca Tabu para um problema de minimização	63
Figura 25 – Operador de cruzamento de ponto único do AG	66
Figura 26 – Operador de cruzamento aritmético do AG	67
Figura 27 – Operador de mutação do AG	67
Figura 28 – Pseudocódigo do Algoritmo Genético	68
Figura 29 – Dinâmica do processo de codificação de áudio MPEG-4-AAC	74
Figura 30 – Simulação do codificador AAC com estrutura de otimização proposta ..	77
Figura 31 – Arquitetura do modelo proposto	78
Figura 32 – Representação da modelagem da métrica <i>ANMR</i>	80
Figura 33 – Codificação do agrupamento de <i>short blocks</i>	84
Figura 34 – Codificação da solução	94
Figura 35 – Codificação do agrupamento das seções	94
Figura 36 – Geração da população inicial para o Algoritmo Genético.....	95
Figura 37 – Fluxograma do processo de cruzamento	98
Figura 38 – Fluxograma do Método de Cruzamento 1	99
Figura 39 – Fluxograma do Método de Cruzamento 2	100
Figura 40 – Primeira rodada do Método de Cruzamento 2	100
Figura 41 – Segunda rodada do Método de Cruzamento 2	101
Figura 42 – Filhos resultantes do Método de Cruzamento 2	101
Figura 43 – Fluxograma de Mutação do Algoritmo Genético	102
Figura 44 – Fluxograma do Método de Mutação 1	103
Figura 45 – Fluxograma do Método de Mutação 2	104
Figura 46 – Fluxograma de geração de vizinhança.....	106
Figura 47 – Fluxograma do Método de Geração 3 de Vizinhança	108
Figura 48 – Estrutura de otimização híbrida para o modelo computacional.....	111

Figura 49 – Comportamento dos <i>scalefactors</i> vs <i>ANMR</i> e taxa de bits	118
Figura 50 – Comportamento dos Códigos de Huffman vs <i>ANMR</i> e taxa de bits.....	121
Figura 51 – Configurações de seções para o experimento de validação	122
Figura 52 – Comportamento do agrupamento dos blocos vs <i>ANMR</i> e taxa de bits	124
Figura 53 – Dinâmica entre <i>scalefactors</i> e <i>scalefactor bands</i> obtida com o codificador de referência (<i>TLS</i>)	132
Figura 54 – Dinâmica entre <i>scalefactors</i> e <i>scalefactor bands</i> obtida com o primeiro experimento do Algoritmo Genético	134
Figura 55 – Dinâmica entre <i>scalefactors</i> e <i>scalefactor bands</i> obtida com o segundo experimento do Algoritmo Genético	136
Figura 56 – Relação entre <i>BT_Max</i> e Lista Tabu para a taxa de 48kbps.....	145
Figura 57 – Relação entre <i>BT_Max</i> e Lista Tabu para a taxa de 116kbps.....	147
Figura 58 – Relação entre <i>BT_Max</i> e Lista Tabu para a taxa de 184kbps.....	149
Figura 59 – Relação entre <i>BT_Max</i> e Lista Tabu para a taxa de 250kbps.....	151
Figura 60 – Evolução do Algoritmo Híbrido em um <i>frame</i> para a taxa de 48kbps...	156
Figura 61 – Evolução do Algoritmo Híbrido em um <i>frame</i> para a taxa de 116kbps.	158
Figura 62 – Evolução do Algoritmo Híbrido em um <i>frame</i> para a taxa de 184kbps.	160
Figura 63 – Evolução do Algoritmo Híbrido em um <i>frame</i> para a taxa de 250kbps.	162

LISTA DE TABELAS

Tabela 1 – Bandas Críticas	38
Tabela 2 – <i>Scalefactors bands</i> para um <i>frame short block</i> para frequência de amostragem de 48kHz	46
Tabela 3 – Parâmetros dos códigos de Huffman para <i>scalefactors</i>	53
Tabela 4 – Parâmetros dos códigos de Huffman para o espectro de frequências	53
Tabela 5 – Níveis para o perfil HE-AAC	56
Tabela 6 – Comparativo trabalhos relacionados	72
Tabela 7 – Interpretação do agrupamento dos <i>short blocks</i> em seções	84
Tabela 8 – Arquivos <i>WAV</i> utilizados nos experimentos	115
Tabela 9 – Análise dos <i>scalefactors vs ANMR</i>	117
Tabela 10 – Análise dos <i>scalefactors vs taxa de bits</i>	118
Tabela 11 – Análise dos Códigos de Huffman <i>vs ANMR</i>	119
Tabela 12 – Análise dos Códigos de Huffman <i>vs taxa de bits</i>	120
Tabela 13 – Análise do agrupamento dos blocos <i>vs ANMR</i>	123
Tabela 14 – Análise do agrupamento dos blocos <i>vs taxa de bits</i>	124
Tabela 15 – Codificação do arquivo de áudio 9 utilizando <i>Two Loop Search</i>	127
Tabela 16 – Relação dos percentuais para a taxa de mutação.....	127
Tabela 17 – Relação dos valores para o Número de Gerações.....	127
Tabela 18 – Resultados do primeiro experimento do AG para a taxa de bits de 48kbps.....	128
Tabela 19 – Resultados do primeiro experimento do AG para a taxa de bits de 116kbps.....	129
Tabela 20 – Resultados do primeiro experimento do AG para a taxa de bits de 184kbps.....	130

Tabela 21 – Resultados do primeiro experimento do AG para a taxa de bits de 250kbps.....	131
Tabela 22 – Média e desvio padrão dos <i>scalefactors</i> obtidos com o codificador de referência (<i>TLS</i>)	133
Tabela 23 – Média e desvio padrão dos <i>scalefactors</i> obtidos com o primeiro experimento do Algoritmo Genético	134
Tabela 24 – Média e desvio padrão dos <i>scalefactors</i> obtidos com o segundo experimento do Algoritmo Genético	136
Tabela 25 – Resultados do segundo experimento do AG para a taxa de bits de 48kbps.....	137
Tabela 26 – Resultados do segundo experimento do AG para a taxa de bits de 116kbps.....	138
Tabela 27 – Resultados do segundo experimento do AG para a taxa de bits de 184kbps.....	139
Tabela 28 – Resultados do segundo experimento do AG para a taxa de bits de 250kbps.....	140
Tabela 29 – Algoritmo Genético vs <i>Two Loop Search</i>	142
Tabela 30 – Relação dos valores de BT_Max.....	143
Tabela 31 – Relação dos valores de tamanho da Lista Tabu	143
Tabela 32 - Resultados da Busca Tabu para a taxa de 48kbps.....	144
Tabela 33 - Resultados da Busca Tabu para a taxa de 116kbps.....	146
Tabela 34 – Resultados da Busca Tabu para a taxa de 184kbps	148
Tabela 35 – Resultados da Busca Tabu para a taxa de 250kbps	150
Tabela 36 – Busca Tabu vs <i>Two Loop Search</i>	152
Tabela 37 – Relação dos valores de BT_Max para o Algoritmo Híbrido	153
Tabela 38 – Relação dos valores de Lista Tabu para o Algoritmo Híbrido.....	153
Tabela 39 – Relação do Número de Gerações para o Algoritmo Híbrido	153
Tabela 40 – Resultados do Algoritmo Híbrido para a taxa de 48kbps.....	154

Tabela 41 – Resultados do Algoritmo Híbrido para a taxa de 116kbps.....	157
Tabela 42 – Resultados do Algoritmo Híbrido para a taxa de 184kbps.....	159
Tabela 43 – Resultados do Algoritmo Híbrido para a taxa de 250kbps.....	161
Tabela 44 – Comparação entre o Algoritmo Híbrido e demais métodos	163
Tabela 45 – Resultados do Algoritmo Híbrido Invertido para a taxa de 48kbps	164
Tabela 46 – Resultados do Algoritmo Híbrido Invertido para a taxa de 116kbps....	165
Tabela 47 – Resultados do Algoritmo Híbrido Invertido para a taxa de 184kbps....	166
Tabela 48 – Resultados do Algoritmo Híbrido Invertido para a taxa de 250kbps....	167
Tabela 49 - Comparação entre o Algoritmo Híbrido Invertido e demais métodos ...	168
Tabela 50 – Tamanho dos arquivos de áudio teste para cada taxa de bits	169
Tabela 51 – Resultados da codificação para a taxa de 48kbps	170
Tabela 52 – Resultados da codificação para a taxa de 116kbps	172
Tabela 53 – Resultados da codificação para a taxa de 184kbps	174
Tabela 54 – Resultados da codificação para a taxa de 250kbps	175
Tabela 55 – Comparação em dB entre as metaheurísticas e a técnica <i>TLS</i> para o Problema de Codificação AAC	177
Tabela 56 – Comparação percentual entre as metaheurísticas e a técnica <i>TLS</i> para o Problema de Codificação AAC	178

SUMÁRIO

1	INTRODUÇÃO	15
1.1	OBJETIVO DO TRABALHO	15
1.2	METODOLOGIA	16
1.3	ORGANIZAÇÃO DO TRABALHO	17
2	ÁUDIO DIGITAL	18
2.1	AMOSTRAGEM DO SINAL	18
2.2	QUANTIZAÇÃO	23
3	CODIFICAÇÃO DE ÁUDIO	26
3.1	HISTÓRICO DA CODIFICAÇÃO DE ÁUDIO	26
3.2	PADRÃO MPEG-4 AAC	28
3.2.1	Modelo Psicoacústico	31
3.2.1.1	Mascaramento	33
3.2.1.2	Bandas Críticas	35
3.2.1.3	Entropia Perceptual	38
3.2.2	Definições dos Blocos	40
3.2.3	MDCT – Modified Discrete Cosine Transform	41
3.2.4	Quantização	45
3.2.5	ScaleFactors	45
3.2.6	Laço Iterativo AAC	48
3.2.7	Codificador de Entropia	51
3.2.8	Predição	54
3.2.9	Perfis e Níveis do MPEG-4 AAC	55
3.3	QUALIDADE PERCEPTUAL DO ÁUDIO	57
4	METAHEURÍSTICAS	60
4.1	BUSCA TABU	60
4.2	ALGORITMO GENÉTICO	64
4.2.1	Codificação do Cromossomo	64
4.2.2	Função Objetivo	65
4.2.3	Operadores de Seleção	65
4.2.4	Operadores de Cruzamento e Mutação	65
5	TRABALHOS RELACIONADOS	69
6	MODELO DE SIMULAÇÃO DO CODIFICADOR MPEG-4 AAC	74

6.1	ARQUITETURA DO MODELO.....	77
6.2	FORMULAÇÃO DO PROBLEMA DE CODIFICAÇÃO AAC	79
6.2.1	Formulação Matemática	83
6.3	ESTRUTURA DE SIMULAÇÃO DO CODIFICADOR MPEG-4 AAC.....	93
6.3.1	Algoritmo Genético para o Problema de Codificação AAC.....	93
6.3.1.1	Codificação do Cromossomo	93
6.3.1.2	População Inicial	95
6.3.1.3	Operador de Seleção	96
6.3.1.4	Operadores de Cruzamento.....	97
6.3.1.5	Operadores de Mutação	102
6.3.1.6	Critério de Parada	104
6.3.2	Algoritmo Busca Tabu para o Problema de Codificação AAC.....	105
6.3.2.1	Codificação da Solução	105
6.3.2.2	Solução Inicial.....	105
6.3.2.3	Estruturas de Vizinhança	105
6.3.2.4	Função Objetivo	109
6.3.2.5	Lista Tabu	109
6.3.2.6	Critério de Aspiração.....	109
6.3.2.7	Critério de Parada.....	110
6.3.3	Algoritmo Híbrido.....	110
7	VALIDAÇÃO E EXPERIMENTOS.....	114
7.1	BIBLIOTECA DE ARQUIVOS DE ÁUDIO TESTE	114
7.2	VALIDAÇÃO.....	116
7.3	EXPERIMENTOS.....	125
7.3.1	Parametrização das Metaheurísticas	126
7.3.1.1	Algoritmo Genético.....	126
7.3.1.2	Busca Tabu	142
7.3.1.3	Algoritmo Híbrido	152
7.3.2	Resultados.....	168
8	CONCLUSÃO	179
	REFERÊNCIAS BIBLIOGRÁFICAS	184

1 INTRODUÇÃO

A possibilidade de troca e acesso ao conteúdo multimídia que hoje nos rodeia, está diretamente relacionada ao avanço das telecomunicações, e também de forma muito intensa, ao avanço dos codificadores de áudio e vídeo. Dados de novembro de 2012 (SANDVINE, 2012), mostram que aproximadamente 58% de todo tráfego da internet fixa nos Estados Unidos é composto de conteúdo multimídia. Para dispositivos móveis este valor é de aproximadamente 38%. No Brasil, estes valores não são tão elevados, mas demonstram a tendência apresentada pelos americanos. Dados de outubro de 2011 (SANDVINE, 2011), mostram que o conteúdo multimídia no Brasil é responsável por aproximadamente 30% do tráfego da internet fixa do país. Desta forma, a eficiência dos codificadores de áudio e vídeo tem uma relação direta em subsidiar e atender este cenário. Assim, através destas ferramentas (CODECs) é possível transportar áudio e vídeo na rede, com uma grande quantidade de informação, de uma maneira mais eficiente, transportando apenas as informações que são fundamentais para distinção do conteúdo.

Para Bauer e Vinton (2006), o conteúdo multimídia tem se destacado ultimamente, principalmente em redes sem fio como WiMAX, Wifi, e telefonia celular (3G/4G), com a utilização de dispositivos móveis. O sucesso destas tecnologias depende da disponibilidade de uma baixa taxa de bits para transportar os conteúdos de áudio e vídeo, objetivo que pode ser alcançado através dos CODECs.

1.1 Objetivo do Trabalho

O codificador de áudio MPEG4-AAC, durante o processo de codificação, deve escolher dinamicamente os parâmetros que determinam a melhor qualidade perceptual do áudio. Este processo de busca dos melhores parâmetros do codificador foi definido como o Problema de Codificação AAC (BAUER e VINTON, 2006). Durante esta etapa, o codificador utiliza um método iterativo conhecido como *TLS (Two Loop Search)* (AGGARWAL, REGUNATHAN e ROSE, 2001), para encontrar os parâmetros para um determinado *frame* de áudio. Porém, este método de busca dos parâmetros utilizado pelo codificador MPEG-4 AAC, não resolve o Problema de Codificação AAC de maneira ótima (BAUER, 2005). Desta forma, este

trabalho propõe um modelo de otimização do codificador AAC baseado em uma solução híbrida de metaheurísticas, de maneira que através de experimentos exaustivos possa se avaliar sua eficiência em relação ao codificador AAC padrão. Desta maneira, o modelo de otimização do codificador AAC será o responsável pela definição dinâmica dos parâmetros de configuração do codificador. Assim, espera-se que a qualidade perceptual do áudio, obtida através do emprego do modelo de otimização proposto, para uma determinada taxa de bits, seja superior quando comparada à qualidade perceptual do codificador AAC em sua estrutura de referência. Além disso, é importante que a estrutura de otimização proposta consiga resolver o problema de codificação AAC, em um tempo computacional considerado aceitável, e ainda avaliar se são permitidas codificações de áudio em tempo real.

A principal contribuição científica deste trabalho é o estudo e desenvolvimento de uma estrutura de otimização para os parâmetros do codificador MPEG-4 AAC, utilizando uma arquitetura híbrida de metaheurísticas. Através desta arquitetura híbrida, busca-se uma qualidade perceptual de áudio superior em relação à solução do codificador de referência, em um tempo computacional adequado para aplicações *off-line*, e se possível aplicações em tempo real.

1.2 Metodologia

O modelo de otimização do codificador MPEG-4 AAC foi desenvolvido utilizando a linguagem de programação C ANSI. Desta forma, esta estrutura de otimização foi inserida no código, também em C, do codificador de referência AAC desenvolvido pelo projeto de software livre FFmpeg (BELLARD), substituindo o método de iteração padrão *TLS* durante a execução dos experimentos de avaliação. Assim, os resultados produzidos pelo codificador com a estrutura proposta puderam ser comparados com os resultados obtidos do codificador com sua estrutura de referência. Estes resultados foram gerados a partir de arquivos de áudio, com diferentes tipos de sons e músicas, de maneira que se pudesse comparar a qualidade perceptual entre as duas estruturas de codificação.

1.3 Organização do Trabalho

Nos capítulos 2 e 3 são abordados o estado da arte, apresentando os conceitos e a forma de operação de um codificador MPEG-4 AAC, como também uma abordagem sobre áudio digital. No capítulo 4 são apresentadas e detalhadas, as metaheurísticas que são utilizadas no desenvolvimento do algoritmo híbrido. No capítulo 5, são descritos alguns trabalhos relacionados, a fim de contextualizar o cenário de otimização do codificador AAC no âmbito da pesquisa operacional. No capítulo 6 é apresentado o modelo do codificador AAC, abordando detalhadamente a modelagem do Problema de Codificação AAC. No capítulo 7 são apresentados os experimentos realizados para a etapa de validação do modelo computacional, para a etapa de parametrização das metaheurísticas, e para os ensaios de avaliação do desempenho do Algoritmo Híbrido, sendo realizada uma análise dos resultados obtidos. Por fim, no capítulo 8 são apresentadas as considerações finais.

2 ÁUDIO DIGITAL

Para que um sinal de áudio possa ser codificado e comprimido por um determinado CODEC de áudio, ele deve estar no formato digital. Entretanto, quando um sinal de áudio é gerado e captado ele está no formato analógico. Desta forma, deve existir um processo de transformação do sinal de áudio analógico, em um sinal de áudio digital.

É interessante observar que para um sinal de áudio no formato analógico, o processo de determinação da quantidade de informação contida, apresenta uma considerável complexidade. Porém, quando o sinal é representado no formato digital, sempre existe uma taxa de bits (*bit rate*) associada à ele, o que permite identificar a quantidade de informação contida no sinal (WATKINSON, 2001).

Algumas etapas devem ser realizadas para que o áudio no formato analógico possa ser representado como um sinal digital. Estes processos em ordem de execução são: amostragem e quantização do sinal (POHLMAN, 1995).

2.1 Amostragem do Sinal

Para que um sinal sonoro seja processado de forma digital ele deve ser adequado aos padrões de trabalho de um sistema digital. Um sinal de áudio quando é captado, está na forma analógica ou contínua. Assim neste formato, antes de ser entregue a um processador de sinais ou CODEC por exemplo, é necessário que o sinal passe por uma etapa denominada de discretização, ou amostragem do sinal (HAYES, 2006). Este processo consiste em representar o sinal através de sua amplitude capturada, a partir de um intervalo de tempo fixo, de forma instantânea. Para ilustrar esta ideia, pode-se observar na Figura 1 que apresenta o comportamento de um sinal contínuo e um sinal amostrado.

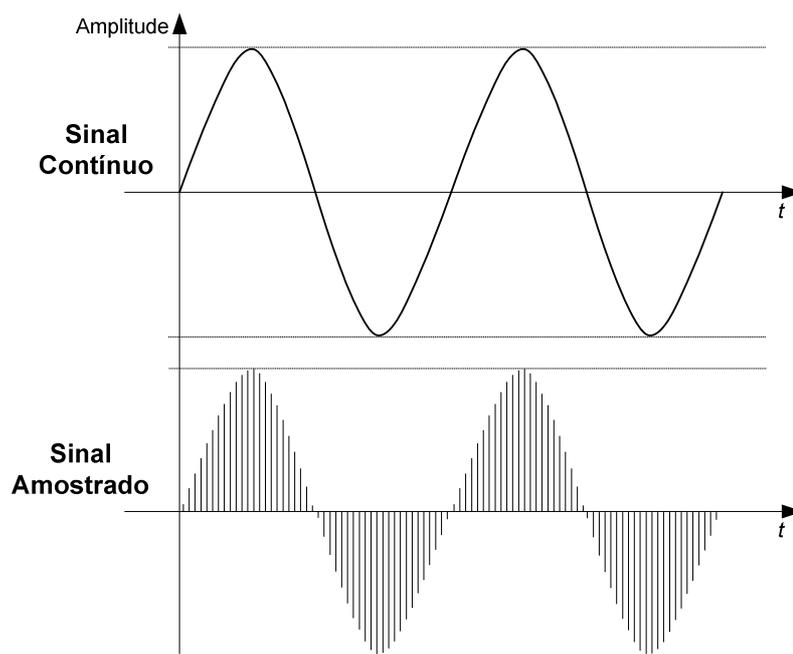


Figura 1 – Sinal contínuo e sinal amostrado

O processo de obtenção de um sinal discreto a partir de um sinal contínuo é realizado através da amostragem periódica, na qual uma sequência de amostras $x[n]$ é obtida a partir de um sinal contínuo $x_c(t)$, conforme a equação (1) (OPPENHEIM e SCHAFER, 1989):

$$x[n] = x_c(nT), \quad -\infty < n < \infty \quad (1)$$

onde, T = período de amostragem, na qual a frequência de amostragem é definida por $f_s = \frac{1}{T}$.

Inicialmente, o sinal contínuo no tempo $x_c(t)$ é multiplicado por uma sequência periódica de impulsos $s(t)$ (OPPENHEIM e SCHAFER, 1989) e definido pela equação (2):

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \quad (2)$$

Desta forma:

$$x_s(t) = x_c(t) \cdot s(t) = \sum_{n=-\infty}^{\infty} x_c(nt) \cdot \delta(t - nT) \quad (3)$$

Após, o sinal amostrado é convertido em um sinal de tempo discreto, no qual os impulsos separados entre si pelo intervalo de amostragem T , são transformados em uma sequência $x[n]$. As amostras assim são indexadas pela variável n , e então representadas pela equação (1).

A fim de representar a equação (3) no domínio da frequência, os sinais $x_c(t)$ e $s(t)$ podem ser representados por $X_c(j\Omega)$ e $S(j\Omega)$, respectivamente. Assim, pode-se definir $S(j\Omega)$ (OPPENHEIM e SCHAFER, 1989) conforme a equação (4):

$$S(j\Omega) = \frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta(\Omega - k\Omega_s) \quad (4)$$

onde $\Omega_s = \frac{2\pi}{T}$ é a frequência de amostragem em radianos por segundo.

Pelo teorema da convolução, uma multiplicação entre duas funções no domínio do tempo, equivale a convolução entre as mesmas duas funções no domínio da frequência (HAYES, 2006). Assim a equação (3) pode ser expressa no domínio da frequência através da equação (5):

$$X_s(j\Omega) = \frac{1}{2\pi} X_c(j\Omega) * S(j\Omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_c(j\Omega - jk\Omega_s) \quad (5)$$

na qual o símbolo $*$ denota a operação de convolução.

Considerando que o sinal $x_c(t)$ é limitado em frequência, então $X_c(j\Omega)=0$ para $|\Omega| > \Omega_0$. Esta consideração pode ser visualizada na Figura 2.

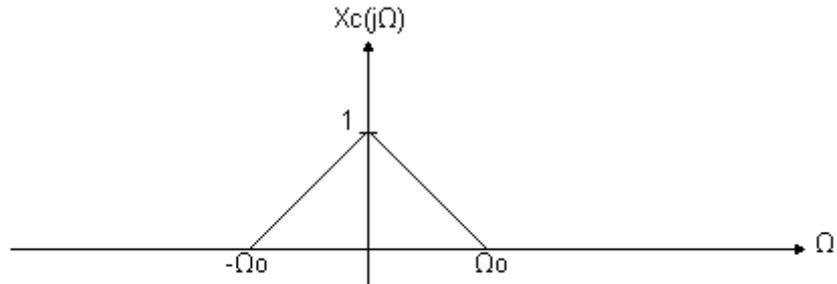


Figura 2 – Faixa de frequência do sinal $x_c(t)$

Fonte: (HAYES, 2006)

Supondo que a sequência de pulsos gerada pela função de amostragem $s(t)$ possua $\Omega_s \geq 2\Omega_0$. Assim, se $s(t)$ for expresso no domínio da frequência como $S(j\Omega)$, então se obterá a Figura 3.

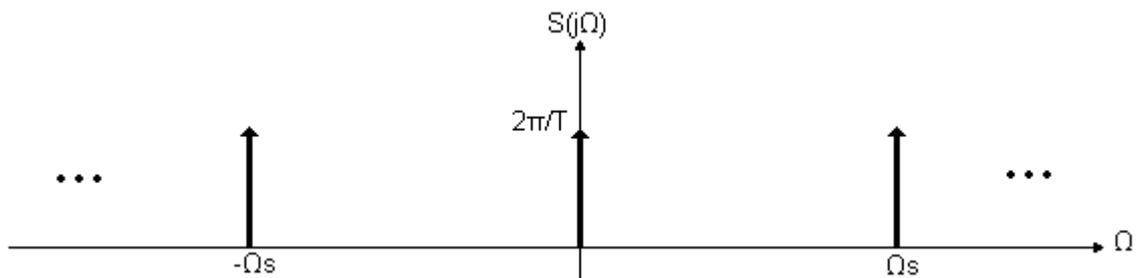


Figura 3 – Espectro da função de amostragem

Fonte: (HAYES, 2006)

Ao convolver a função de amostragem $S(j\Omega)$, com a função do sinal $X_c(j\Omega)$, no domínio da frequência, é gerado o sinal amostrado $X_s(j\Omega)$ (HAYES, 2006) (Figura 4).

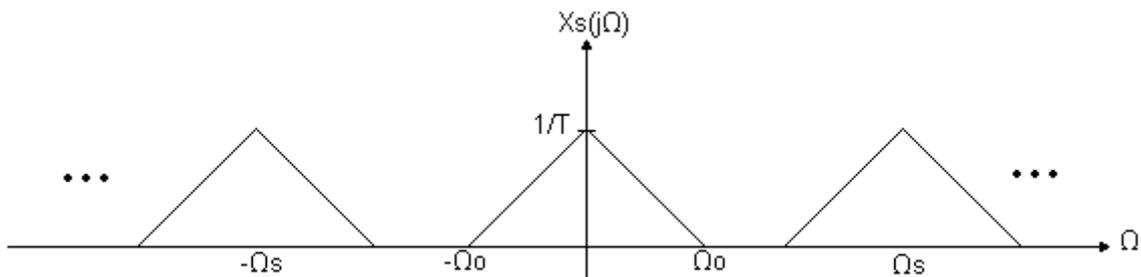


Figura 4 – Espectro do sinal amostrado com $\Omega_s > 2\Omega_0$

Fonte: (HAYES, 2006)

Entretanto, se $X_c(j\Omega)$ é amostrado com $\Omega_s < 2\Omega_0$, os espectros deslocados $X_c(j\Omega - jk\Omega_s)$ são sobrepostos e somados, gerando uma função $X_s(j\Omega)$ que apresenta uma deformação (Figura 5).

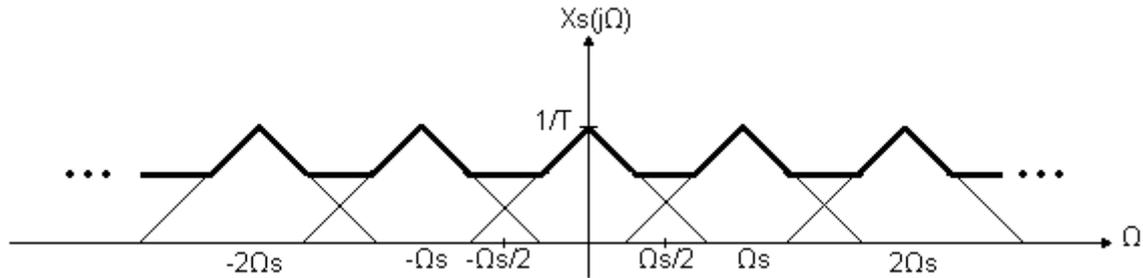


Figura 5 – Espectro do sinal amostrado com $\Omega_s < 2\Omega_0$

Fonte: (HAYES, 2006)

A sobreposição das componentes espectrais apresentada na Figura 5 é chamada de *aliasing*. Quando este fenômeno ocorre com o sinal $x_c(t)$ por exemplo, seu conteúdo de frequência é corrompido, não sendo mais possível recuperar a informação de $X_c(j\Omega)$ a partir de $X_s(j\Omega)$ (HAYES, 2006).

Na situação da Figura 4, em que $\Omega_s \geq 2\Omega_0$, não ocorre o fenômeno *aliasing*, e o sinal $x_c(t)$ pode ter sua informação recuperada por meio de um filtro-passa baixas, sem qualquer deformação. Desta maneira pode-se definir o teorema de amostragem: um sinal contínuo no tempo $x_c(t)$ que possui sua largura de banda limitada em B Hz, pode (na teoria) ser unicamente recuperado a partir das amostras $x_c[nT]$, onde $f_s = \frac{1}{T}$, desde que $f_s > 2B$ (CARTINHOOR, 2000).

Assim, sendo Ω_0 a máxima frequência que se deseja representar após um processo de discretização, a frequência de amostragem Ω_s deve ser:

$$\Omega_s = \frac{2\pi}{T} \geq 2\Omega_0 \quad (6)$$

onde T = período de amostragem.

O teorema de amostragem, equação (6), é conhecido também como teorema de Nyquist, pois foi introduzido por Harry Nyquist. Por este motivo, também se chama a mínima frequência de amostragem necessária para representar um determinado sinal, de frequência de Nyquist. Desta forma, a definição da frequência

de amostragem é determinante no projeto de um sistema de áudio ou comunicações, pois é neste ponto que inicia o tratamento do sinal no mundo digital.

2.2 Quantização

Dentro do processo de digitalização de um sinal, depois de realizada a amostragem é necessário efetuar a quantização do sinal. Isso significa que para cada ponto do sinal amostrado, se deverá atribuir um valor binário, para que sua amplitude possa ser representada no formato digital.

A ideia é dividir o eixo de amplitude em regiões (Figura 6), sendo que cada região seja representada por um valor binário. Assim, cada ponto amostrado é enquadrado dentro da região que mais se aproxima do seu valor de amplitude. Deve-se observar que quanto maior a quantidade de regiões, mais resolução é inserida à etapa de quantização, e mais fidedignos estarão os valores quantizados. Assim, os valores quantizados são somente uma aproximação dos valores reais (POHLMAN, 1995).

Na Figura 6 é representado um sinal analógico devidamente amostrado, atendendo ao teorema de Nyquist, e o mesmo sinal quantizado. Observa-se que existem oito valores possíveis para se representar a amplitude, pois neste exemplo foram utilizados três bits ($2^3 = 8$ valores distintos de amplitude). Assim, para cada ponto amostrado é atribuído o valor binário que melhor representa este valor de amplitude.

É interessante observar que a etapa de amostragem já insere uma perda de informação ao sinal analógico, e de maneira análoga quando quantizado, o sinal mais uma vez tem a sua informação reduzida (POHLMAN, 1995). Isso se deve ao fato de que a quantidade de pontos, ou informação, contida em um sinal analógico é infinita. Desta forma, quando aplicadas as etapas de amostragem e quantização ao sinal, elas por utilizarem processos com uma quantidade finita de valores, acabam tornando finita também, a quantidade de informação contida no sinal digital. Essa diferença de informação entre o sinal analógico e digital representa a inserção de um determinado erro sobre o sinal digital. Esse erro é denominado como erro de quantização.

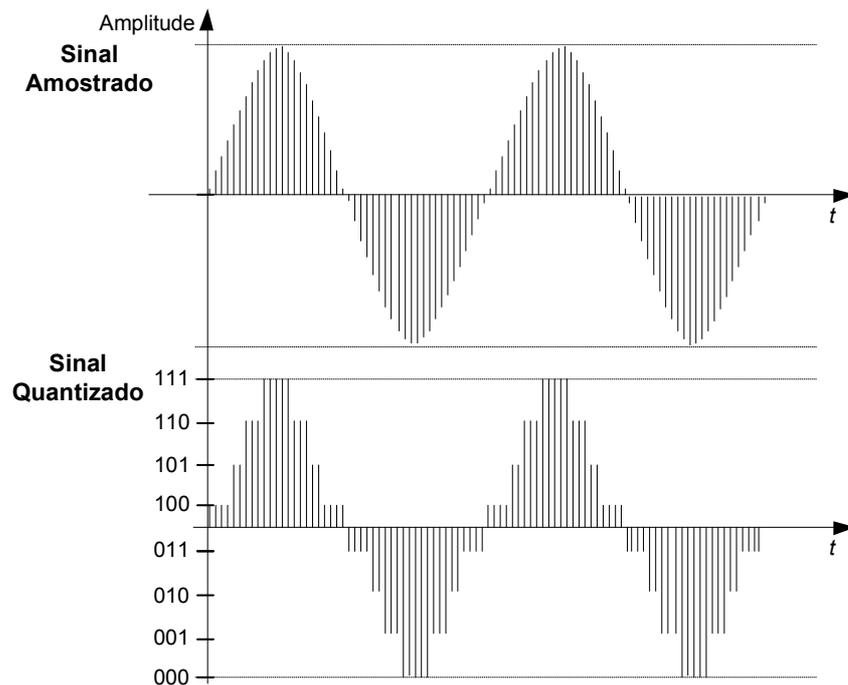


Figura 6 – Sinal Quantizado

O erro de quantização é medido através do *SRE* (*Signal-to-Error Ratio*) através dos níveis de tensão do Sinal (*S*) e do Erro (*E*) (POHLMAN, 1995). Assim, considerando *N* o número de intervalos do processo de quantização e *n* o número de bits, tem-se a equação (7):

$$N = 2^n \quad (7)$$

Uma metade dos *N* valores disponíveis para o processo de quantização é utilizada para valores maiores que zero da forma de onda, e a outra metade para valores menores que zero. Sendo *Q* o intervalo de quantização, o máximo valor de pico que o nível de sinal pode excursionar é de $\pm Q2^{n-1}$. Assumindo um sinal senoidal como entrada, o valor máximo em *rms* (*root mean square*) é S_{rms} , expresso pela equação (8) (POHLMAN, 1995):

$$S_{rms} = \frac{Q2^{n-1}}{\sqrt{2}} \quad (8)$$

De forma análoga, a energia do erro de quantização também pode ser modelada matematicamente.

A quantização pode ser dita uniforme e não uniforme. O erro de quantização é independente e identicamente e uniformemente distribuído entre os limites $+Q/2$ e $-Q/2$, onde Q é um intervalo de quantização. Assim, a função densidade de probabilidade que representa estes eventos, é uma função uniforme com amplitude de $1/Q$. Desta forma, o erro de quantização *rms* (E_{rms}), pode ser encontrado integrando o produto do erro e sua probabilidade, conforme a equação (9) (POHLMAN, 1995):

$$E_{rms} = \sqrt{\int_{-\infty}^{\infty} e^2 p(e) de} = \left[\frac{1}{Q} \int_{-\frac{Q}{2}}^{+\frac{Q}{2}} e^2 de \right]^{\frac{1}{2}} = \left[\frac{Q^2}{12} \right]^{\frac{1}{2}} = \frac{Q}{\sqrt{12}} \quad (9)$$

Desta forma, a relação entre o sinal e o erro de quantização é definida pela equação (10):

$$\frac{S}{E} = \left[\frac{S_{rms}}{E_{rms}} \right]^2 = \frac{\frac{Q2^{n-1}}{\sqrt{2}}}{\frac{Q}{\sqrt{12}}} = \frac{3}{2} (2^{2n}) \quad (10)$$

Expressando esta relação em decibéis, obtém-se a equação (11):

$$\frac{S}{E} (dB) = 10 \log \left[\frac{3}{2} (2^{2n}) \right] = 20 \log \left[\left(\frac{3}{2} \right)^{\frac{1}{2}} (2^n) \right] = 6.02n + 1.76 \quad (11)$$

A partir desta relação, pode-se verificar que para cada bit que se adiciona no processo de quantização, a relação S/E aumenta, o que provoca também um aumento na quantidade de informação, um decréscimo de 6dB no erro de quantização, e conseqüentemente uma melhoria na qualidade do áudio. Para 16 bits de quantização por exemplo, a relação S/E é aproximadamente 98dB, enquanto que para 15 bits de quantização a relação é de 92dB.

A relação S/E que foi modelada matematicamente é também conhecida como ruído de quantização (POHLMAN, 1995). Desta forma, é correto afirmar que o aumento do ruído de quantização ocorre com o decréscimo da quantidade de bits empregada no processo de quantização.

3 CODIFICAÇÃO DE ÁUDIO

O processo de codificação de áudio é uma transformação aplicada ao sinal de áudio digital, a fim de representá-lo de uma nova maneira, facilitando a sua manipulação num processo de transmissão, armazenamento, criptografia, etc. (POHLMAN, 1995).

Associado ao processo de codificação está o mecanismo de compressão do sinal. A compressão de um sinal de áudio pode ser classificada como uma compressão com perdas ou sem perdas (POHLMAN, 1995). Ao se optar por uma compressão sem perdas, o sinal de áudio será compactado, diminuindo a quantidade de bits necessários em sua representação, e poderá ser recuperado obtendo a forma do sinal original. Porém se a compressão for com perdas, ao se diminuir a quantidade de bits em sua representação, o sinal não poderá mais atingir sua forma original anterior à codificação (WATKINSON, 2001).

Desta forma, nesta seção é abordado o processo de codificação de áudio, fazendo um breve histórico deste cenário e, na sequência, um detalhamento profundo sobre o funcionamento e as características do codificador de áudio MPEG-4 AAC. São abordadas as diversas estratégias utilizadas no processo de codificação, que permitem com que o codificador AAC consiga uma boa compressão e ao mesmo tempo mantenha a qualidade perceptual.

3.1 Histórico da Codificação de Áudio

A codificação de áudio teve seus primeiros estudos em 1988 com o codificador MASCAM, desenvolvido pelo instituto de tecnologia alemão *Institut für Rundfunktechnik* (IRT), para o sistema de Difusão de Áudio Digital (DAB). A partir deste formato o codificador MUSICAM (*Masking pattern adapted Universal Subband Integrated Coding And Multiplexing*), em 1989, foi desenvolvido pelo IRT em conjunto com o CCETT (*Centre Commun d'études de Télévision et Télécommunications*), *Philips* e *Matsushita Electric Industrial*. Estes dois codificadores têm como principal característica a divisão do sinal em um grande número de sub-bandas. Assim, cada banda é tratada e codificada de uma maneira diferente, de acordo com sua relevância perceptual (FISCHER, 2010).

Ao mesmo tempo em que os métodos citados eram estudados, foi desenvolvido o ASPEC (*Adaptive Spectral Perceptual Entropy Coding*) pela parceria das empresas *Fraunhofer Gesellschaft* e *Thomson*. Este método utiliza a DCT (*Discrete Cosine Transform*) para obter o sinal no domínio da frequência, e elimina as componentes que são irrelevantes (FISCHER, 2010).

Em 1991, o padrão MPEG-1 (ISO/IEC 11172-3) de compressão de áudio incorporou os dois métodos de codificadores MUSICAM e ASPEC, criando um novo codificador. O MPEG-1 compreende três camadas (*layers*):

Layer I – Baixa complexidade na estrutura do codificador e baixa compressão;

Layer II – Média complexidade na estrutura do codificador;

Layer III – Alta complexidade na estrutura do codificador e alta compressão, sendo baseado na codificação em sub-bandas e transformada;

O MPEG-1 *Layer III*, é conhecido popularmente como o MP3, e até o desenvolvimento do AAC, foi o codificador mais amplamente utilizado (BRANDENBURG, 1999). É interessante observar que a estrutura do codificador MP3, baseia-se na união das melhores características do MUSICAM e ASPEC.

Em 1994 foi criado o codificador de áudio MPEG 2 (ISO/IEC 13818-3), em que apenas o *Layer II* foi estendido com a inserção da codificação de áudio multicanal (5.1).

Paralelamente aos codificadores de áudio do grupo MPEG, a Dolby Labs desenvolveu em 1990, o padrão de codificação *Dolby Digital Audio*, que também ficou conhecido como AC-3 Audio. Este padrão foi muito difundido no cinema, e hoje em dia é adotado como padrão em muitos filmes, como também no formato de TV Digital americano (ATSC). Do ponto de vista de qualidade de áudio, praticamente não há diferença entre MPEG Audio e Dolby Digital (FISCHER, 2010).

O codificador AAC surgiu em sua primeira versão, ainda no MPEG-2 Audio em 1997, sob a norma ISO/IEC 13818-7, trazendo muitas melhorias sobre seu antecessor MP3. Devido utilizar uma estrutura mais flexível e modular, e implementar novas estratégias de codificação, o codificador conseguiu obter melhores resultados de compressão.

Na sequência em 1999, ele foi incorporado ao MPEG-4 Audio com o título de AAC plus (ISO/IEC 14496-3), trazendo novamente novas estratégias e algoritmos de codificação, melhorando ainda mais a sua performance de compressão. Em 2003 e 2004, o codificador apresentou a implementação de novas funcionalidades, e então foi chamado de HE-AACv1 e HE-AACv2 (*High Efficiency – Advanced Audio Coding*).

O último padrão de 2004 se mantém até hoje, e este formato tem sido muito utilizado em diversos dispositivos. Alguns exemplos são: iPhone, Playstation 3, Xbox 360. Em 2003 este padrão foi o adotado pelo sistema de TV Digital do Japão ISDB-T. Conseqüentemente em 2007 o padrão de TV Digital brasileiro (ISDB-Tb), que é totalmente baseado no sistema japonês, também passou a utilizar o codificador de áudio MPEG-4 HE-AAC.

O processo de codificação de áudio é padronizado por dois grupos internacionais de normatização, ISO/IEC MPEG Audio (*Moving Pictures Experts Group*) e ITU (*International Telecommunication Union*). O primeiro efetivamente atua mais na normatização de codificadores de áudio e vídeo, enquanto que o segundo, além de codificadores, atua na recomendação e normatização de sistemas de telecomunicações (FISCHER, 2010).

3.2 Padrão MPEG-4 AAC

O padrão MPEG4 AAC especifica um codificador que é considerado como um codificador de áudio perceptual. Isso significa que uma de suas principais estratégias de compressão é procurar eliminar elementos de redundâncias e informações irrelevantes, que não são perceptíveis ao ouvido humano (PAINTER e SPAN, 2000). Na Figura 7 é apresentado o diagrama em blocos do codificador de áudio MPEG-4 AAC, de maneira que cada bloco é detalhado a seguir (ISO/IEC, 2005):

- Sinal de Entrada: deve ser um sinal de áudio sem compressão, codificado apenas em *PCM*, formato utilizado nos arquivos *WAV* e nos *CDs*;
- Controle de Ganho: correção realizada sobre o sinal de áudio de entrada, aplicando-se diferentes ganhos para diferentes bandas de frequência do sinal. Este processo é efetuado para que as amplitudes do sinal de entrada

sejam niveladas antes da conversão do sinal do domínio do tempo para o domínio da frequência;

- MDCT: através da técnica da transformada discreta do cosseno modificado, o sinal de entrada é convertido para o domínio da frequência;
- TNS (*Temporal Noise Shaping*): esta técnica modela o ruído presente no sinal de áudio de alguns *frames* escolhidos, dinamicamente no tempo. Desta forma, o codificador consegue determinar em quais faixas de frequência a relação sinal ruído é baixa, e compensar este efeito através de ganho nesta faixa;
- M/S Estéreo e *Intensity Coupling*: ambas são técnicas de compressão para sinal de áudio estéreo. O método *Middle/Side* (M/S Estéreo) para diminuir a quantidade de informação, trabalha com a soma e subtração dos canais R e L, formando respectivamente os canais M e S. A técnica *Intensity Coupling* é utilizada apenas nas bandas de frequência mais elevadas, acima de 6kHz (ISO/IEC, 2004). Este método, para determinadas faixas de frequência, propõe a representação do sinal estéreo através da soma da energia dos canais L e R. As bandas são definidas baseadas na informação do modelo psicoacústico;
- Predição: técnica que procura prever o sinal de áudio a partir de seu comportamento e dinâmica atual;
- *ScaleFactor*: ganho aplicado às componentes de frequência, de maneira que define a importância de cada coeficiente para a qualidade perceptual do sinal de áudio;
- Quantização: etapa que efetivamente determina a quantidade de bits alocada para cada componente espectral a ser codificada;
- Codificador de Entropia: bloco que determina a correlação entre os coeficientes quantizados e realiza uma codificação para diminuir a quantidade de dados na transmissão e armazenamento;
- Controle da taxa de bits e distorção: laço de iteração que procura determinar a melhor combinação entre os parâmetros do codificador, de maneira que obtenha uma qualidade perceptual para o áudio com a taxa de bits especificada;

- Modelo Psicoacústico: modelo matemático que representa o comportamento do ouvido humano e permite que o codificador possa avaliar sua codificação baseada na qualidade perceptual obtida do sinal de áudio;
- Formatação do fluxo de dados: nesta etapa o fluxo de dados do áudio codificado é multiplexado junto com as informações auxiliares necessárias para a decodificação do sinal;
- Sinal de Saída: sinal de áudio comprimido no formato AAC.

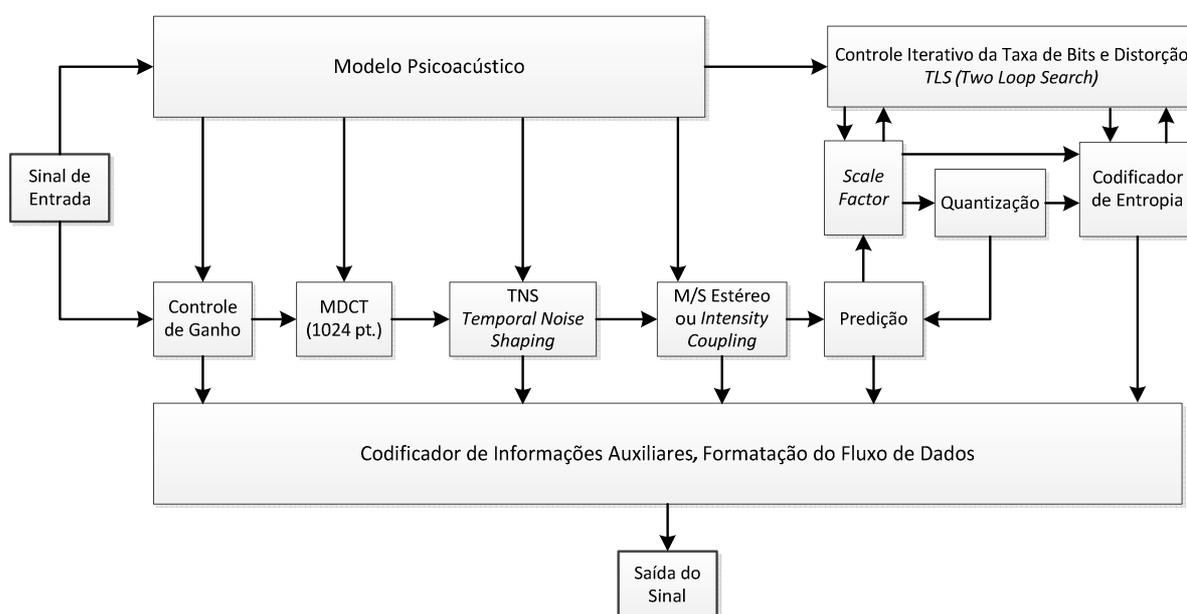


Figura 7 – Diagrama do Codificador MPEG-4 AAC

Fonte: (ISO/IEC, 2005) – Tradução do próprio autor

Pode-se observar que o codificador utiliza diversas estratégias em conjunto, para que ao final do processo seja obtido um sinal de áudio comprimido, com uma baixa taxa de bits. Entretanto, se o áudio codificado não apresentar uma qualidade perceptual, o codificador não obteve êxito na tarefa que lhe foi designada. Desta forma, deve existir um compromisso intrínseco à estrutura do codificador, de diminuir a taxa de bits mantendo a qualidade perceptual do áudio.

O processo de codificação de áudio utilizando o padrão MPEG-4 AAC pode ser categorizado como uma compressão de dados com perdas. Isso significa que depois de submetido à compressão, o sinal jamais será o mesmo, pois parte da informação contida no sinal foi descartada. Porém, existem processos de codificação

de sinais de áudio caracterizados pela compressão sem perdas. Assim o sinal de áudio após ser codificado, diminui a quantidade necessária de bits para sua representação. Porém no processo de decodificação, o sinal é recuperado de forma íntegra, ou seja, o sinal decodificado será exatamente igual ao sinal fonte utilizado no codificador. Um exemplo de codificador de áudio sem perdas que pode ser citado é o FLAC (*Free Lossless Audio Codec*) (MORIYA e IWAKAMI, 2000).

3.2.1 Modelo Psicoacústico

Para que os codificadores de áudio pudessem cada vez mais melhorar seu desempenho, ou seja, aumentar a compressão de dados mantendo uma qualidade adequada de áudio, foi necessário integrar cada vez mais técnicas e estratégias ao codificador. Uma destas técnicas foi a inserção de um modelo psicoacústico dentro do codificador (FASTL e ZWICKER, 2007). Este modelo tem o objetivo de trazer ao codificador, informações sobre como o ouvido humano responde aos sons, relacionando as propriedades físicas dos mesmos (que podem ser medidas cientificamente de forma objetiva), com as respostas fisiológicas e psicológicas por elas provocadas. Desta forma, a psicoacústica é essencialmente o estudo da percepção do som, e procura traduzir ao algoritmo do codificador, a maneira como o ouvido humano funciona, permitindo um controle sobre a experiência auditiva das pessoas.

Conhecendo o comportamento do ouvido humano, o codificador é então capaz de determinar quais informações contidas no sinal de áudio tem mais relevância em proporcionar uma qualidade perceptual sonora para as pessoas. A partir desta definição, o codificador determina quais informações do sinal sonoro podem ser desprezadas e quais devem ser mantidas. Esta análise dos sons proposta pela psicoacústica é realizada no domínio do tempo e também no domínio da frequência.

O ouvido é capaz de identificar frequências pertencentes à faixa de 20Hz a 20kHz (POHLMAN, 1995). Desta forma, caso o codificador utilizasse apenas esta informação do modelo psicoacústico, eliminaria do sinal de áudio todas as frequências fora da banda especificada. De forma análoga, em relação ao nível ou

amplitude do sinal de áudio, existe um nível mínimo de pressão sonora, $20\mu\text{Pa}$, que é necessário para que os seres humanos consigam diferenciá-lo do silêncio (FASTL e ZWICKER, 2007). Quanto ao limite máximo, em acústica se trabalha com o nível de pressão sonora na faixa do limiar doloroso do ouvido, que é aproximadamente 100 Pa . Uma maneira de representar o nível de pressão sonora de um sinal de áudio é através da relação *SPL* (*Sound Pressure Level*), expressa em decibéis (dB). Esta medida é calculada sempre em relação ao menor nível de pressão sonora que os seres humanos são capazes de ouvir e possui um comportamento logarítmico, conforme a equação (12) (PAINTER e SPAN, 2000).

$$SPL = 10 \log\left(\frac{P}{2 \cdot 10^{-5}}\right) (dB) \quad (12)$$

onde P é a pressão sonora de um determinado sinal que se deseja obter o valor da relação *SPL*.

A equação (12) descreve o nível de pressão sonora em relação ao mínimo nível de pressão necessário para um ouvido humano ser capaz de identificá-lo, porém não descreve o comportamento do ouvido. Para isso, é necessária uma expressão que relacione a pressão sonora *SPL* do sinal de áudio, com a sua frequência. Assim, considerando o ouvido de uma pessoa jovem com uma audição apurada, a curva do limiar de audição humana pode ser expressa pela equação (13) (PAINTER e SPAN, 2000):

$$T_q(f) = 3,64 \left(\frac{f}{1000}\right)^{-0,8} - 6,5e^{-0,6\left(\frac{f}{1000} - 3,3\right)^2} + 10^{-3} \left(\frac{f}{1000}\right)^4 \quad (13)$$

Assim, $T_q(f)$ representa o limiar absoluto de audição do ouvido humano. Quando aplicado à compressão de sinais de áudio, $T_q(f)$ pode ser interpretado como o máximo nível de energia permitido de distorção, que o processo de

codificação pode introduzir no domínio da frequência de maneira que mantenha a qualidade perceptual do áudio (PAINTER e SPAN, 2000).

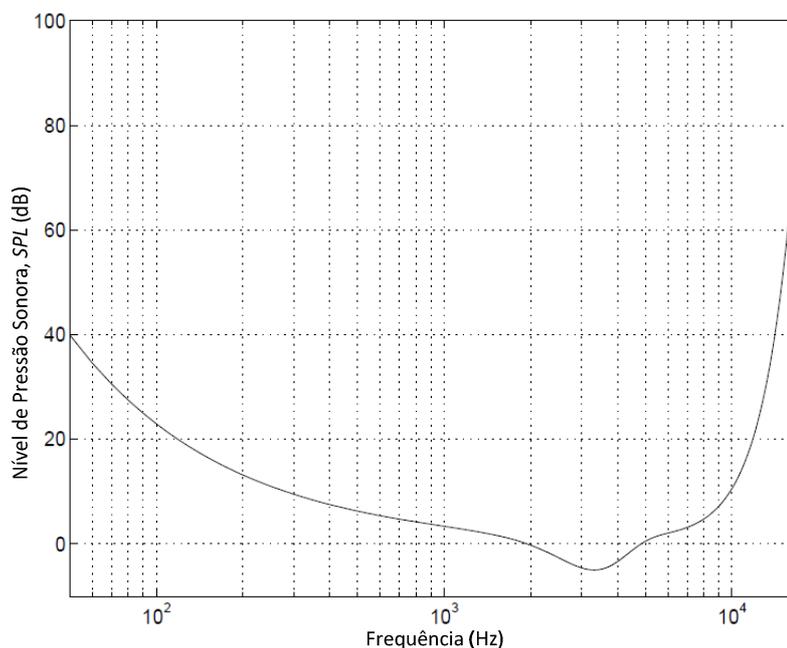


Figura 8 – Limiar absoluto de audição do ouvido humano

Fonte: (PAINTER e SPAN, 2000) – Tradução do próprio autor

A Figura 8 representa a curva de $T_q(f)$, e desta forma é possível verificar esta relação entre o nível mínimo de pressão sonora necessário para o ouvido humano distinguir um sinal sonoro, e a sua frequência.

3.2.1.1 Mascaramento

Mascaramento é um fenômeno que ocorre de um sinal de áudio sobre outro. Ele pode ser identificado no domínio da frequência e no domínio do tempo (FASTL e ZWICKER, 2007). Assim, a curva de resposta do ouvido humano (Figura 8) é alterada dinamicamente em função dos sinais de áudio presentes.

Na Figura 9 existem duas componentes representadas no domínio da frequência, A e B, com frequências de 500Hz e 200Hz respectivamente. A amplitude do sinal B é superior à amplitude de A, e desta forma, causa uma distorção na curva

de sensibilidade do ouvido humano, mascarando o sinal A. Sem a presença do sinal B, o sinal A seria ouvido normalmente, porém devido ao sinal B, o sinal A é mascarado e desta maneira não é possível ouvi-lo.

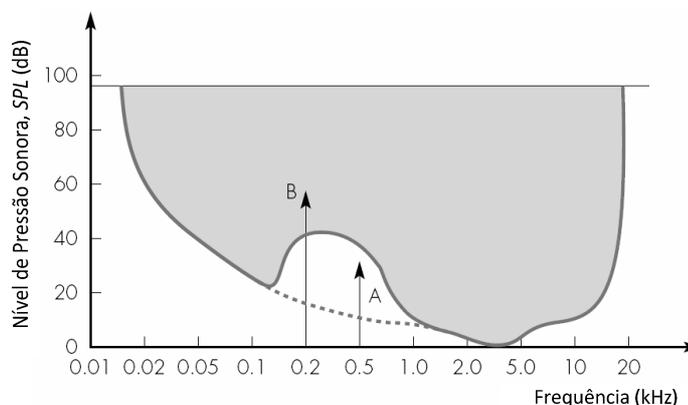


Figura 9 – Efeito de mascaramento no domínio da frequência

Fonte: (POHLMAN, 1995) – Tradução do próprio autor

Na seção 3.2.3 será abordado o tema *MDCT*, um método matemático baseado nos conceitos de *Fourier* que permite representar um sinal no domínio da frequência a partir de um sinal no domínio do tempo. Desta forma, a magnitude das componentes de frequência que compõem o sinal de áudio em análise pode ser determinada. De posse destas magnitudes das componentes de frequência é possível identificar quais frequências serão mascaradas, e não precisam ser codificadas ou transmitidas pelo codificador (POHLMAN, 1995). Assim, são codificadas apenas as componentes de frequência que estão acima da curva do limiar absoluto do ouvido humano visto na Figura 8.

De forma semelhante, no domínio do tempo o efeito de mascaramento também é verificado. Quando um som com um nível mais elevado é emitido, o ouvido demora alguns milissegundos (aproximadamente 50ms) para ouvir outro som novamente (POHLMAN, 1995). Este retardo de tempo, apresentado pelo ouvido, provoca um mascaramento temporal nos sons que sucedem o sinal principal. Na Figura 10, este mascaramento é demarcado pela região abaixo da curva. Desta forma, os sinais no tempo, que acontecerem dentro desta região, não são

codificados e portanto desprezados pelo codificador de áudio (PAINTER e SPAN, 2000).

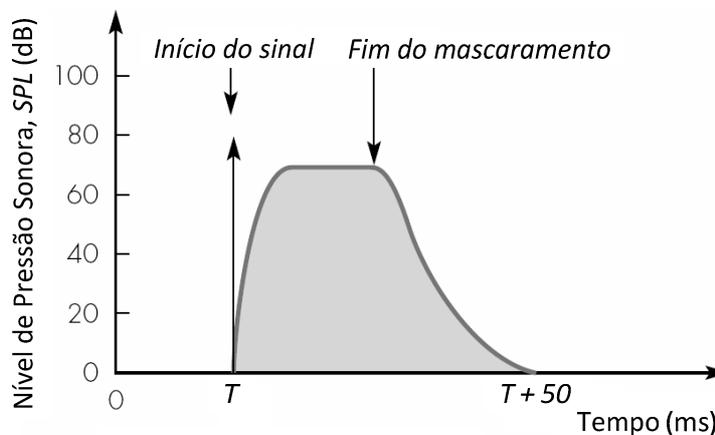


Figura 10 – Efeito de mascaramento no domínio do tempo

Fonte: (POHLMAN, 1995) – Tradução do próprio autor

Através do mascaramento espectral e temporal operando de maneira conjunta, o modelo psicoacústico que existe dentro da estrutura do codificador, funciona auxiliando outras etapas e estratégias do mesmo, no sentido de avaliar quais informações do sinal de áudio devem ser mantidas, e quais devem ser eliminadas (POHLMAN, 1995). Desta forma, codificadores que são caracterizados como perceptuais, como é o caso do MPEG-4 AAC, devem possuir dentro de sua estrutura um modelo psicoacústico (KANDADAI, 2007).

3.2.1.2 Bandas Críticas

Utilizar o limiar absoluto de audição (Figura 8) para moldar o espectro de distorções no processo de codificação, representa o primeiro passo para a codificação perceptual. Novos elementos são incorporados ao modelo psicoacústico no momento em se estuda como o ouvido humano faz uma análise espectral (PAINTER e SPAN, 2000).

Um sinal sonoro ao entrar no canal do ouvido é conduzido até o ouvido interno, onde a membrana basilar detecta a amplitude e frequência do som. Essas

vibrações são convertidas em impulsos elétricos, e conduzidas até o cérebro como informações de áudio, pelas fibras nervosas (POHLMAN, 1995). A membrana basilar possui aproximadamente 30.000 células ciliares, que são responsáveis pela identificação e reconhecimento de frequência. Porém elas respondem a estímulos mais fortes em sua região local, e para cada frequência, há um ponto da membrana basilar em que a vibração é máxima. O tamanho desta região é proporcional ao logaritmo da frequência do som, e ao seu redor haverá uma faixa, de cerca de 1,5 mm, onde a vibração estará presente e será atenuada conforme se afasta da região. O conceito das bandas críticas então surge, com a identificação destas regiões, devido haver comportamentos distintos para distintas regiões da membrana basilar.

Uma maneira de determinar uma banda crítica é apresentada na Figura 11. São usados dois sinais de áudio de diferentes frequências (S1 e S2) e uma pequena banda de ruído, inserida exatamente ao centro dos sinais. Os sinais S1 e S2 mascaram o ruído, de maneira que não é possível escutá-lo. Porém, os sinais S1 e S2 têm suas frequências afastadas pela diferença Δf , que é incrementada suavemente. Mantido sempre ao centro dos sinais S1 e S2, haverá uma distância Δf em que será possível ouvir o ruído. Neste ponto, para a frequência central do ruído, a distância Δf será igual à largura de banda da banda crítica (FASTL e ZWICKER, 2007).

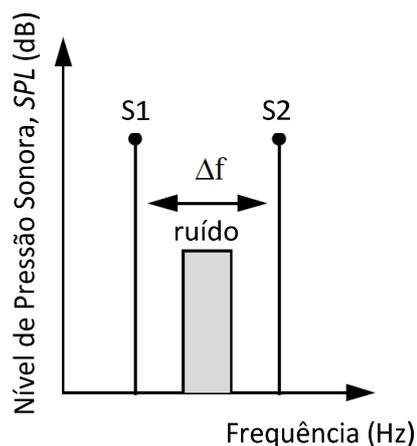


Figura 11 – Análise de banda crítica

Fonte: (PAINTER e SPAN, 2000) – Tradução do próprio autor

Deve-se observar que, quando dois sinais se situam dentro de uma banda crítica, o de maior energia poderá dominar a percepção e mascarar o outro estímulo sonoro. Portanto, dependendo dos níveis, dois tons distintos somente serão distinguidos um do outro, quando estiverem em bandas críticas diferentes (FASTL e ZWICKER, 2007).

Assim uma banda crítica define uma faixa em torno de uma frequência central, a qual está associada a uma região da membrana basilar. Desta forma, para cada região da membrana é possível definir uma banda crítica (POHLMAN, 1995).

A largura da faixa das bandas críticas corresponde a um espaçamento uniforme de 1,5 mm ao longo da membrana basilar, o que corresponde a aproximadamente 100 Hz para frequências abaixo de 500 Hz, e de aproximadamente 20% da frequência central da banda para frequências acima de 500Hz. Portanto, a resposta de amplitude em frequência, para cada banda crítica, pode ser modelada como um filtro passa-faixas com largura de faixa crescente com a frequência (PAINTER e SPAN, 2000).

A distância de uma banda crítica é referenciada na literatura como “um Bark”. Assim a faixa de frequências de áudio é dividida em 25 bandas críticas e representada em uma nova escala. A equação (14) define a relação entre frequência e a escala Bark:

$$Bark = 13 \arctan\left(\frac{0,76f}{1000}\right) + 3,5 \arctan\left(\frac{f}{7500}\right)^2 \quad (14)$$

Os valores das frequências centrais e os limites superiores e inferiores de cada banda crítica são apresentados na Tabela 1:

Nº Banda Crítica (Bark)	Frequência Central (Hz)	Banda Crítica (Hz)	Limite Inferior (Hz)	Limite Superior (Hz)
0	50	-	-	100
1	150	100	100	200
2	250	100	200	300
3	350	100	300	400
4	450	110	400	510
5	570	120	510	630
6	700	140	630	770
7	840	150	770	920
8	1000	160	920	1080
9	1170	190	1080	1270
10	1370	210	1270	1480
11	1600	240	1480	1720
12	1850	280	1720	2000
13	2150	320	2000	2320
14	2500	380	2320	2700
15	2900	450	2700	3150
16	3400	550	3150	3700
17	4000	700	3700	4400
18	4800	900	4400	5300
19	5800	1100	5300	6400
20	7000	1300	6400	7700
21	8500	1800	7700	9500
22	10500	2500	9500	12000
23	13500	3500	12000	15500
24	18775	6550	15500	22050

Tabela 1 – Bandas Críticas

Fonte: (FASTL e ZWICKER, 2007) – Tradução do próprio autor

3.2.1.3 Entropia Perceptual

Johnston dos Laboratórios Bell, em 1988, uniu os conceitos de mascaramento psicoacústico com os princípios de quantização de sinais, e definiu a Entropia Perceptual (*PE*) (JOHNSTON, 1988). *PE* é definida como a medida de informação perceptual relevante contida em um sinal de áudio. Expressa em bits por amostra, *PE* representa o limite teórico de compressibilidade de um sinal de áudio (BAUER, 2005). Em outras palavras, a métrica *PE* estima o limiar que representa o nível máximo de ruído, que pode ser gerado pelo processo de codificação, de forma que seja inaudível quando adicionado ao sinal de áudio de entrada (JOHNSTON, 1988).

Para calcular *PE*, inicialmente o sinal de áudio é dividido no tempo em *frames*, e cada um destes é convertido para domínio da frequência. Os limiares de mascaramento são então calculados com base nas regras do modelo psicoacústico

descrito nas seções anteriores deste capítulo. Esta análise é realizada unindo-se o conceito de bandas críticas e mascaramento espectral/temporal. Finalmente, a PE é calculada como o número de bits necessários para reconstruir o áudio sem inserção de ruído perceptível (KANDADAI, 2007).

Na equação (15) é possível verificar a definição matemática da Entropia Perceptual (JOHNSTON, 1988):

$$PE = \frac{1}{M} \sum_{i=1}^N \sum_{b_i=bl_i}^{bh_i} \left[\log_2 \left(2 \left[\frac{Re(t_{b_i})}{\sqrt{\frac{6}{w_i k_i}}} \right] + 1 \right) + \log_2 \left(2 \left[\frac{Im(t_{b_i})}{\sqrt{\frac{6}{w_i k_i}}} \right] + 1 \right) \right] (bits/amostra) \quad (15)$$

onde:

i = índice da banda crítica,

bh_i = limite superior da banda i ,

bl_i = limite inferior da banda i ,

k_i = número de componentes da transformada da banda i ,

b_i = índice de componentes da transformada da i -ésima banda crítica,

w_i = é o inverso da máscara do limite de audição dos seres humanos,

$Im(t_{b_i})$, $Re(t_{b_i})$ = parte imaginária e real, respectivamente, dos coeficientes da transformada da i -ésima banda crítica,

M = número de coeficientes quantizados.

Desta forma, é possível verificar que sinais de áudio de baixa Entropia Perceptual quando codificados, podem ser reduzidos de forma mais eficiente. Porém os sinais de alta entropia necessitam de uma quantidade maior de informação a ser transportada, que não deve ser reduzida (POHLMAN, 1995). Na verdade a ideia dos codificadores perceptuais é justamente tentar manter constante a Entropia Perceptual do sinal de áudio codificado.

3.2.2 Definições dos Blocos

O codificador de áudio MPEG-4 AAC, como outros codificadores perceptuais, divide o sinal no tempo em fatias denominadas de *frames*. Assim cada *frame* é composto por um determinado número de amostras, que determinam o seu tamanho. Esta divisão é realizada para que cada *frame* possa ter um tratamento individual dentro do processo de codificação. Essa manipulação do sinal em partes se faz necessária, devido ao sinal de áudio variar intensamente no tempo, de maneira que suas características possam ser mais bem representadas e analisadas através de uma sequência de *frames* (BRANDENBURG, 1999). A Figura 12 ilustra um sinal de áudio dividido em partes iguais no tempo (*frames*):

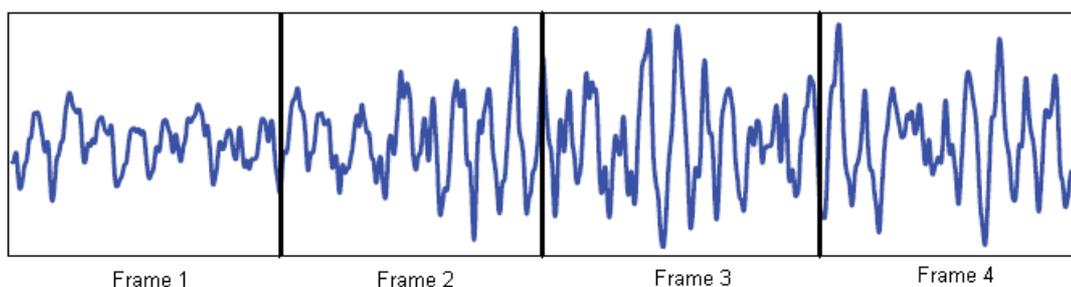


Figura 12 - Divisão de um sinal de áudio dinâmico em frames

Os *frames* podem ser divididos em blocos. Os blocos do codificador AAC podem conter 256 ou 2048 amostras, sendo caracterizados como *short blocks* e *long blocks* respectivamente. Para uma frequência de amostragem de 48 kHz, um *short block*, representa o sinal de áudio durante um tempo de 5,3 ms, enquanto que um *long block* representa 42,7 ms cada bloco.

A decisão do codificador sobre qual bloco utilizar, *short* ou *long*, é baseada no modelo psicoacústico, mais precisamente na métrica Entropia Perceptual (*PE*). Assim, para um *frame* do sinal de áudio de baixa entropia, ou seja, um pequeno valor para *PE*, o *frame* é codificado utilizando um *long block*. Neste caso, *PE* está indicando que não existem muitas variações dentro do *frame*, e assim não há necessidade de utilizar uma quantidade expressiva de bits em sua representação (BAUER, 2005). Porém, se outro *frame* do sinal de áudio é indicado pelo índice *PE* como de entropia elevada, então ele deverá ser codificado utilizando uma sequência

de 8 *short blocks*. Assim, devido à alta variação do sinal de áudio dentro do *frame*, para melhor representá-lo, deve-se utilizar blocos menores do sinal a fim de representar seus detalhes de maneira mais rica. Deve-se observar que o tamanho do conjunto dos 8 *short blocks*, equivalem ao tamanho de um *long block*. Assim, os 8 *short blocks* possuem a mesma duração de tempo que um *long block*, porém a quantidade de bits utilizada para representar os 8 *short blocks* é maior do que a quantidade de bits utilizada em um *long block*.

3.2.3 MDCT – *Modified Discrete Cosine Transform*

Depois do sinal de áudio ter sido dividido no tempo em *frames*, estes são convertidos do domínio do tempo para o domínio da frequência. Isso se faz necessário, pois uma análise espectral permite diferenciar e interpretar melhor as características dinâmicas e estáticas do sinal de áudio (POHLMAN, 1995). O método matemático utilizado para isso é a *MDCT (Modified Discrete Cosine Transform)*, que é uma variação da *DCT (Discrete Cosine Transform)* que por sua vez baseia-se nos conceitos de *Fourier* (OPPENHEIM e SCHAFER, 1989).

A fim de evitar descontinuidades no sinal de áudio, a *MDCT* baseia-se em uma utilização de *frames* sobrepostos no tempo. Este processo é conhecido como a técnica de *overlapping*, e intrínseca a ela esta associada outra técnica chamada de janelamento dos *frames* (NIAMUT, 2006).

O processo de janelamento de sinais no domínio do tempo é realizado multiplicando-se a função que descreve a janela que se deseja utilizar, pelo *frame* do sinal de áudio desejado. Existem diversas funções de janelamento, porém no codificador MPEG-4 AAC são utilizadas, a janela do Seno representada pela equação (16) e a janela de *KBD (Kaiser-Bessel Derived)* representada pelas equações (17) e (18) (ISO/IEC, 2005).

$$w_s(n) = \sin \left[\frac{\pi}{2N} \left(n + \frac{1}{2} \right) \right], \text{ para } n \ 0 \leq n \leq 2N - 1 \quad (16)$$

$$w_k(n) = \frac{I_0\left(\pi\alpha\sqrt{1 - \left(\frac{2n}{2N-1}\right)^2}\right)}{I_0(\pi\alpha)} \text{ para } n \ 0 \leq n \leq 2N-1 \quad (17)$$

$$I_0(n) = \sum_{l=0}^{\infty} \left(\frac{\left(\frac{x}{2}\right)^k}{k!}\right)^2 \quad (18)$$

onde:

α = determina o molde da janela, valores maiores produzem uma janela mais estreita.

Desta forma, o sinal de áudio $x_{i,n}$ contido no *frame* i , após aplicada a função de janelamento $w(n)$, pode ser expresso por $z_{i,n}$, conforme a equação (19) (OPPENHEIM e SCHAFER, 1989):

$$z_{i,n} = w(n) \cdot x_{i,n} \quad (19)$$

A Figura 13 apresenta a janela Seno gerada através do MATLAB, de maneira que permite visualizar a forma que o sinal de áudio será submetido depois de realizada a etapa de janelamento.

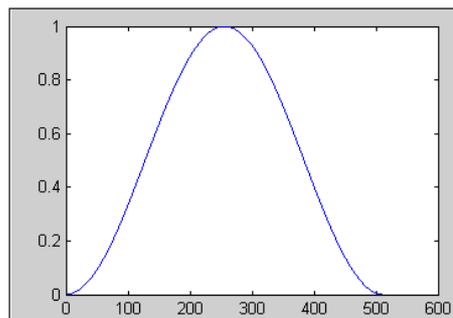


Figura 13 – Função de janelamento Seno

O efeito provocado sobre o sinal $x_{i,n}$ pelo janelamento da função $w(n)$, detalhado na Figura 13, pode ser visualizado na Figura 14. É interessante observar, que o sinal do *frame* que foi submetido ao janelamento, sofreu atenuações em suas

amplitudes iniciais e finais. Assim, o processo de sobreposição dos *frames* adjacentes tem suas emendas suavizadas.

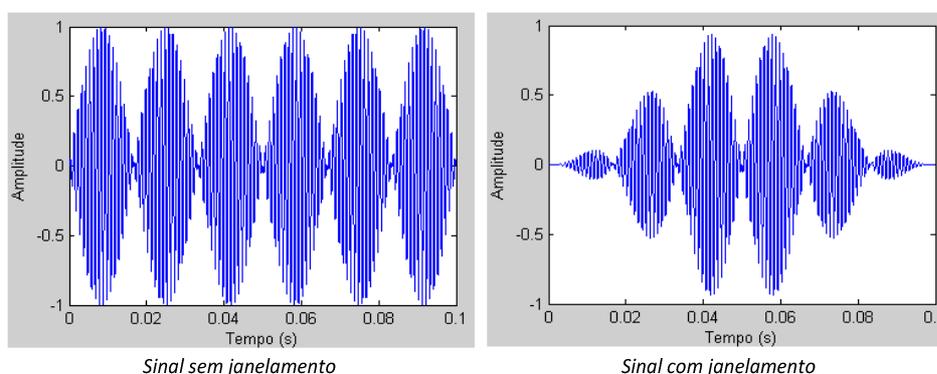


Figura 14 – Sinal amostrado e o mesmo multiplicado pela janela Seno

O percentual de sobreposição de *frames* que é padronizado no MPEG-4 AAC é de 50% (ISO/IEC, 2005). Isso significa, que das 2048 amostras contidas nos *long blocks*, e das 256 contidas nos *short blocks* vistos na seção anterior, apenas metade destes valores são novas amostras. As demais amostras são do bloco adjacente anterior, com o qual as amostras são sobrepostas.

Durante o processo de codificação, os blocos podem ser alternados dinamicamente entre *long blocks* e *short blocks*, de acordo com a métrica da Entropia Perceptual. Para que seja feita uma alteração de *long blocks* para *short blocks* e vice versa, são necessários outros dois tipos de blocos, denominados de blocos de transições (ISO/IEC, 2005). O bloco de transição entre um *long block* e um *short block*, é o *long start block*. De maneira análoga, o bloco de transição entre um *short block* e um *long block*, é o *long stop block* (ISO/IEC, 2005). Estes blocos de transição são necessários para adequar a sobreposição, já que não é possível efetuar este procedimento entre um *long* e *short block* diretamente (Figura 15).

Depois de efetuado o processo de janelamento, é possível determinar os coeficientes X_k da *MDCT* do sinal $z_{i,n}$ utilizando a equação (20):

$$X_k = \sum_{n=0}^{2N-1} z_n \cdot \cos \left[\frac{\pi}{N} \left(n + \frac{1}{2} + \frac{N}{2} \right) \left(k + \frac{1}{2} \right) \right] \quad (20)$$

onde:

z_n = são as amostras do sinal de áudio de um frame devidamente janelado pela função Seno ou *KBD*;

k = é o índice das amostras no domínio da frequência;

n = é o índice das amostras no domínio do tempo;

N = quantidade de amostras em um *long block* ou 8 *short blocks* (2048);

Através da equação (19), verifica-se que no domínio do tempo são utilizadas 2048 amostras, para gerar 1024 coeficientes no domínio da frequência. Desta forma, as 2048 amostras de um *frame* são utilizadas apenas como entrada para a transformada *MDCT*, que entrega ao final do processo 1024 componentes de frequência. Deve-se ressaltar que esta quantidade de coeficientes representam, para a faixa de frequências de áudio, uma resolução de aproximadamente 20Hz (WATKINSON, 2001).

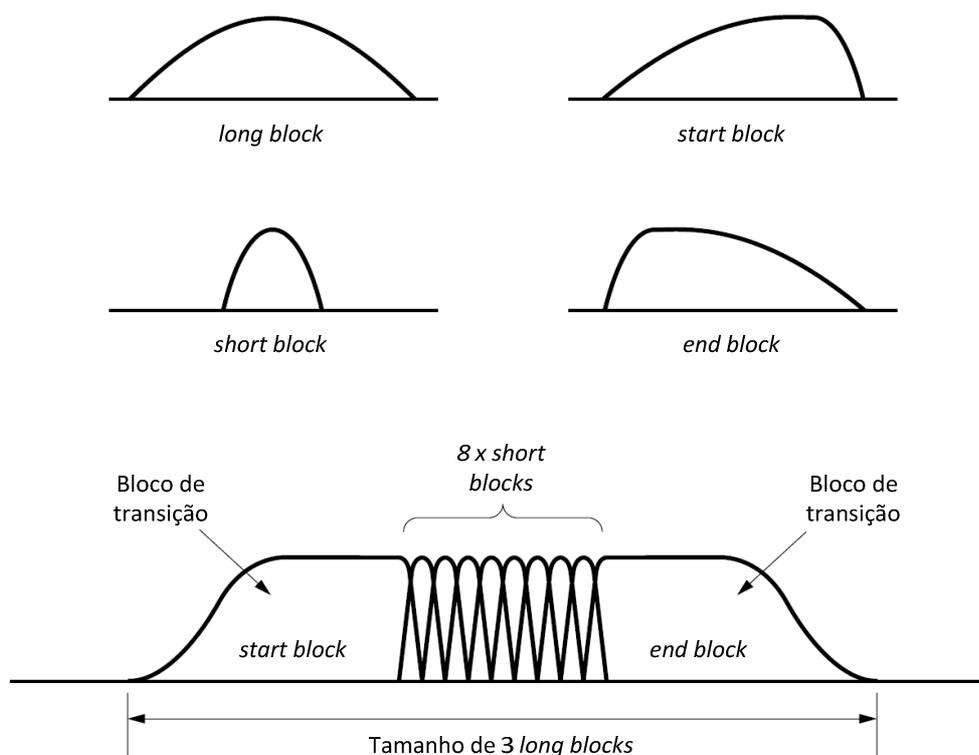


Figura 15 – Estrutura dos blocos no MPEG-4 AAC

Fonte: (WATKINSON, 2001) – Tradução do próprio autor

3.2.4 Quantização

Depois de obtidas as componentes de frequência de um bloco, utilizando a *MDCT*, é necessário realizar o processo de quantização. Na seção 2.2 este processo foi abordado, mas no domínio tempo, quantizando cada ponto amostrado da forma da onda do sinal de áudio. Nesta etapa, o objetivo é conseguir uma representação das componentes espectrais da maneira mais compacta possível, e ao mesmo tempo introduzir a menor distorção perceptível ao sinal de áudio (HERRE, 1999). Desta forma, deseja-se uma baixa taxa de bits e ao mesmo tempo uma qualidade perceptual do sinal sonoro.

3.2.5 ScaleFactors

O processo de definição que determina a quantidade de bits necessários para cada componente espectral é definida utilizando o modelo psicoacústico. O método básico para ajustar o ruído de quantização no domínio da frequência é modelar o ruído (*Noise Shaping*) utilizando fatores de escala (*scalefactors*) (BAUER, 2005). Assim, um *scalefactor* representa um valor de ganho que é usado para alterar a amplitude de um conjunto de coeficientes espectrais adjacentes. Este conjunto formado por coeficientes é denominado de *scalefactor bands*, ou seja, uma banda inteira de frequência que é escalonada por um *scalefactor*. Desta maneira, a efetiva resolução do processo de quantização é controlada pelo *scalefactor* (ISO/IEC, 2005).

A largura das *scalefactor bands* é construída baseada nas bandas críticas do sistema auditivo humano, visto na seção 3.2.1.2. Por esse motivo a quantidade de *scalefactor bands* em um espectro, como também a sua largura, depende do tamanho do bloco analisado pela *MDCT* e da frequência de amostragem (ISO/IEC, 2004). Desta forma, cada *scalefactor band* possui um *scalefactor* associado, que representa um determinado ganho, que deve ser aplicado para todos os coeficientes espectrais contidos na *scalefactor band*. Na Tabela 2 é possível visualizar a divisão de um *short block* em *scalefactors bands* para uma frequência de amostragem de 48kHz.

ScaleFactors Band	Deslocamento das componentes de frequência (MDCT)
0	0
1	4
2	8
3	12
4	16
5	20
6	28
7	36
8	44
9	56
10	68
11	80
12	96
13	112
14	128

Tabela 2 – *Scalefactors bands* para um *frame short block* para frequência de amostragem de 48kHz

Fonte: (ISO/IEC, 2005) – Tradução do próprio autor

Com o intuito de aumentar a compressão, os *scalefactors* associados à *scalefactors band* que possuem somente valores zerados, são ignorados no processo de codificação e conseqüentemente não são transmitidos. Cada intervalo ou passo do *scalefactor* durante a quantização representa um salto de 1.5dB, o que representa 1.41 vezes (ISO/IEC, 2004).

A informação do ganho que o *scalefactor* aplica aos coeficientes de uma determinada *scalefactor band*, também deve ser transmitida junto com os dados codificados para o decodificador. Estes dados são denominados dados auxiliares e servem para auxiliar o decodificador no processo de remontagem do sinal de áudio (BAUER, 2005). Assim cada *scalefactor* representa uma informação adicional, que deve ser inserida ao feixe de dados do codificador AAC (HERRE, 1999). Desta forma, a taxa total de bits do sinal codificado é formada pela soma dos bits necessários para representar os coeficientes espectrais quantizados, com os bits necessários para representar os dados auxiliares.

Um caso particular para as *scalefactor bands* é quando o *frame* manipulado é do tipo *short block*. Como este bloco sempre é utilizado em conjunto, num total de 8

blocos, as *scalefactor bands* destes blocos, podem ser agrupadas em seções por semelhança (ISO/IEC, 2004). A ideia por trás do agrupamento é que os *frames* adjacentes de um grupo de 8 *short blocks*, possam compartilhar os mesmos *scalefactors*, para todas as *scalefactor bands* de uma mesma seção. Com esta operação, se reduz a quantidade de *scalefactors* que devem ser transmitidos pelo codificador, e assim, também a quantidade de dados auxiliares necessários. Com a diminuição das informações auxiliares, a taxa total de bits é reduzida, e assim o codificador consegue obter uma melhoria em seu desempenho (ISO/IEC, 2004).

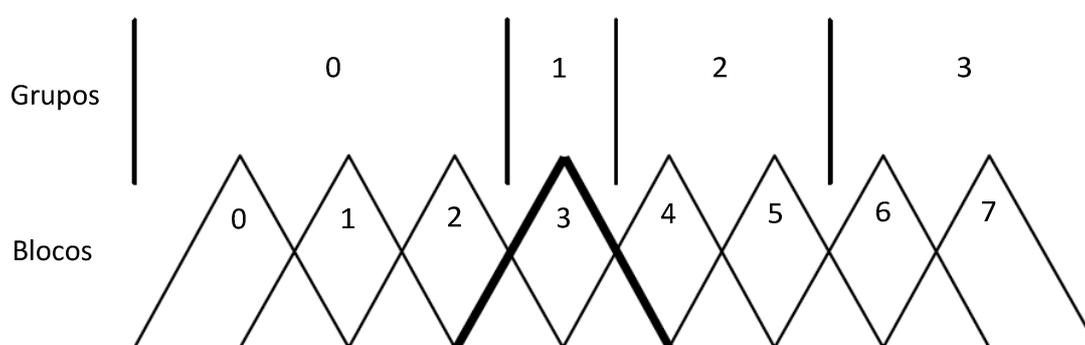


Figura 16 – Exemplo de agrupamento de 8 *short blocks*

Fonte: (ISO/IEC, 2004) – Tradução do próprio autor

A Figura 16 apresenta um exemplo de agrupamento para os 8 *short blocks*. Identifica-se que os 8 blocos são codificados como 4 grupos ou seções. Na seção 0 existem 3 blocos, na seção 1 apenas um bloco, e nas seções 2 e 3, existem 2 blocos para cada. Desta forma, os *scalefactors* dos *short blocks* 0, 1 e 2, pertencentes ao grupo zero, são os mesmos para cada uma das *scalefactors bands*. Isso ocorre também para os demais grupos. É interessante observar que existem $2^7 = 128$ possibilidades de agrupamento das seções (BAUER, FELLERS e DAVIDSON, 2006).

Os valores de todos os *scalefactors* são definidos dentro dos laços de quantização. Depois disso, os *scalefactors* são codificados como dados auxiliares. A codificação dos *scalefactors* é realizada de maneira diferencial, ou seja, apenas é codificado o valor da diferença entre o *scalefactor* atual e o seu adjacente anterior. Por exemplo, se o *scalefactor* $sf(2)$ possui o valor 17, e o $sf(3)$ o valor 19, então o

valor do *scalefactor* $sf(3)$ que será codificado, representará a diferença entre eles, -2 (menos dois) (ISO/IEC, 2005).

3.2.6 Laço Iterativo AAC

Para que a taxa de bits do sinal de saída do codificador, não ultrapasse o valor estipulado, e não seja inserido qualquer ruído perceptivo, existe um controle iterativo dentro do codificador (HOFFMANN, 2002). Este processo possui dois *loops*, um denominado de iteração do laço externo, responsável por controlar a distorção produzida pela quantização, e o outro de iteração do laço interno que controla a taxa de bits da quantização (PAINTER e SPAN, 2000). Na literatura este método também é conhecido como *Two Loop Search (TLS)*. Na Figura 17 é apresentada a estrutura de iteração que controla a taxa de bits e a quantidade de distorção do codificador MPEG-4 AAC.

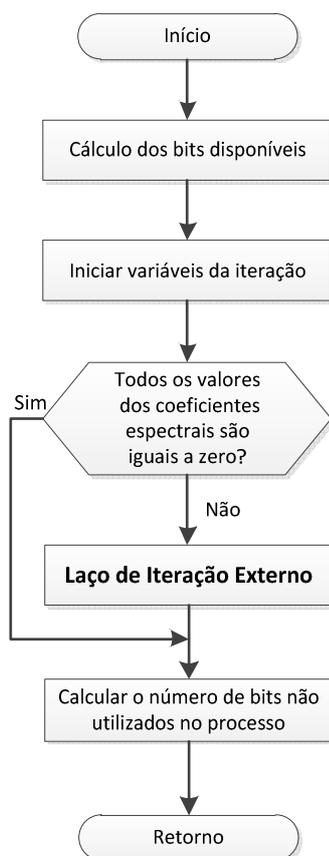


Figura 17 – Laço de Iteração AAC

Fonte: (ISO/IEC, 2004) – Tradução do próprio autor

Efetivamente, os laços responsáveis pelo controle do codificador são os laços de interação interno e externo, e são detalhados a seguir:

- **Iteração do laço Interno:** Esta iteração efetivamente realiza a quantização das componentes de frequência, e chama o processo de codificação do sem ruído (*Noiseless Coding*). Após, o laço conta a quantidade de bits utilizados na codificação dos coeficientes e dos *scalefactors*. Se a quantidade de bits contabilizada for superior ao máximo valor pré-estabelecido de *bit rate* (taxa de bits), o processo de quantização é refeito. Este laço se repete até que o valor obtido para a taxa de bits, não exceda o máximo valor (HOFFMANN, 2002). O fluxograma do laço interno é apresentado na Figura 18.

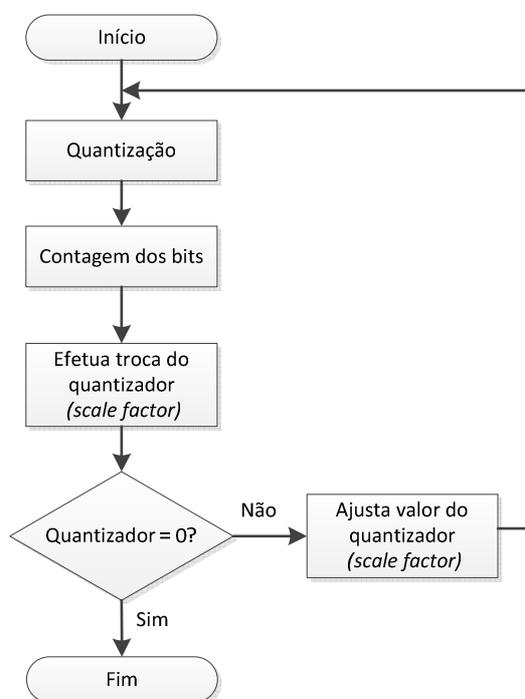


Figura 18 – Iteração do laço interno

Fonte: (ISO/IEC, 2004) – Tradução do próprio autor

- **Iteração do laço Externo:** A iteração inicia executando o laço interno. Assim os valores de distorção são calculados para cada *scalefactors band* (ISO/IEC, 2004). Se algum valor de distorção for superior ao máximo permitido pelo modelo psicoacústico para alguma *scalefactor band*, os valores dos *scalefactors* são

ajustados e a iteração do laço interno é chamada novamente (ISO/IEC, 2004). Na Figura 19 é detalhado o fluxograma do laço externo.

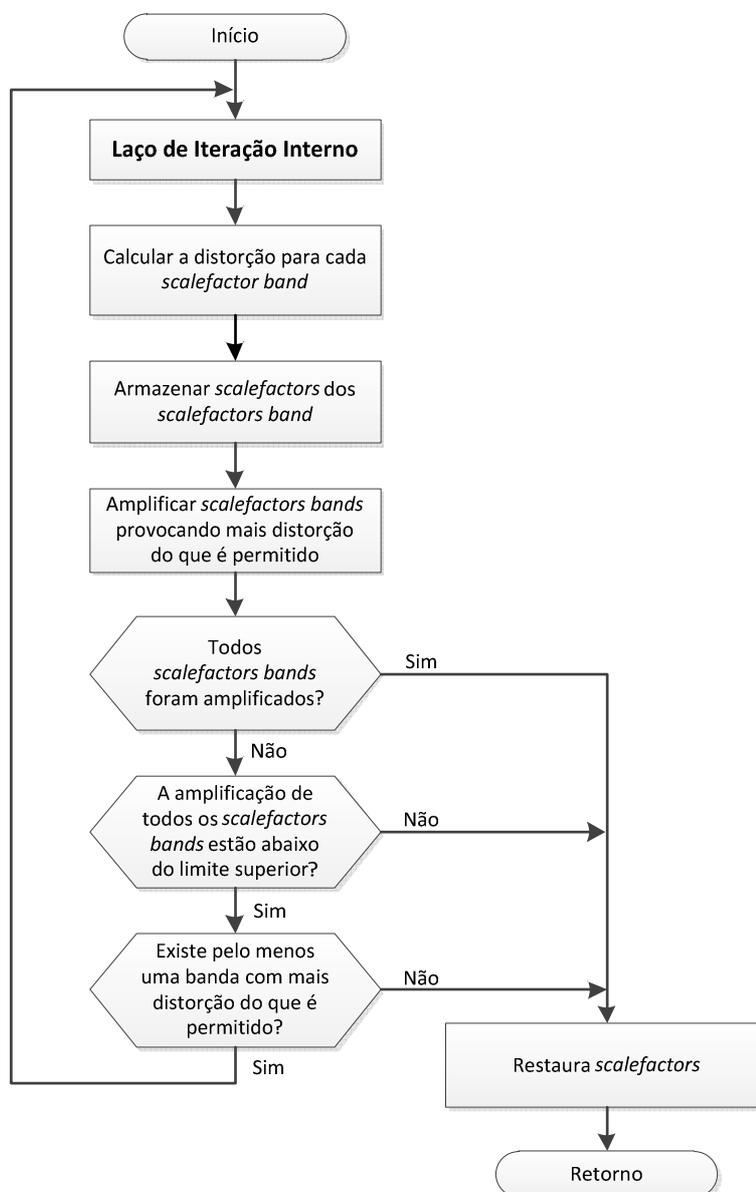


Figura 19 – Iteração do laço externo

Fonte: (ISO/IEC, 2004) – Tradução do próprio autor

Com o objetivo de evitar laços infinitos é necessário definir uma condição de saída para o laço externo. Isso se faz necessário, pois enquanto o laço interno reduz a quantidade de bits no processo de quantização, para não exceder a taxa máxima, o laço externo aumenta a quantidade de bits, pois a distorção está elevada. Esta dinâmica entre os laços, se não há um ponto de parada, conduz o algoritmo para infinitas iterações (HOFFMANN, 2002). Desta forma, o laço é repetido uma

determinada quantidade de vezes, limitando a quantidade de ajustes dos *scalefactors*. Este critério de parada aceita alguma quantidade de distorção adicionada ao sinal codificado, de maneira que este problema não é resolvido de maneira ótima (BAUER, 2005).

3.2.7 Codificador de Entropia

Após os coeficientes espectrais terem sido quantizados, utilizando um determinado *scalefactor*, ambos são associados em um único dado, e passam a constituir um símbolo (STOLFI, 2008). Assim, este símbolo é codificado por um codificador de entropia antes de ser inserido no fluxo de dados de saída do codificador. Um codificador de entropia explora a não uniformidade da distribuição de probabilidade dos dados quantizados (símbolos), para gerar a menor taxa de bits que representa uma dada sequência de símbolos (DUTOIT e MARQUÉS, 2009). Este processo é realizado atribuindo uma quantidade maior de bits para símbolos menos prováveis, e para símbolos mais prováveis, uma quantidade reduzida de bits. A base teórica que envolve este estudo é denominada de Teoria da Informação (COVER e THOMAS, 2006).

A entropia mede a incerteza de um determinado evento, ou seja, quanto maior a entropia, mais incerteza está associada ao evento e, portanto, mais informações são necessárias para representá-lo em um processo de transmissão. Pode ser provado que, para uma fonte de símbolos equiprováveis a entropia é máxima, isto é, se todos os símbolos têm a mesma probabilidade de ocorrência, a incerteza sobre qual símbolo será produzido pela fonte é máxima (COVER e THOMAS, 2006).

No codificador de áudio MPEG-4 AAC, este processo de análise da entropia dos dados é realizada utilizando os Códigos de Huffman (*Huffman Code Book - HCB*) (ISO/IEC, 2005). Para gerar os Códigos de Huffman, é necessário um conhecimento prévio da função densidade de probabilidade, da entidade fonte dos eventos a serem codificados. No caso do codificador de áudio, estes eventos são os dados dos coeficientes espectrais quantizados e dos valores dos *scalefactors*. O algoritmo que gera os códigos utiliza um processo iterativo, unindo os eventos de

menor probabilidade a cada iteração, até que todos os eventos estejam conectados uns aos outros. A Figura 20 apresenta um exemplo de um processo de construção dos Códigos de Huffman:

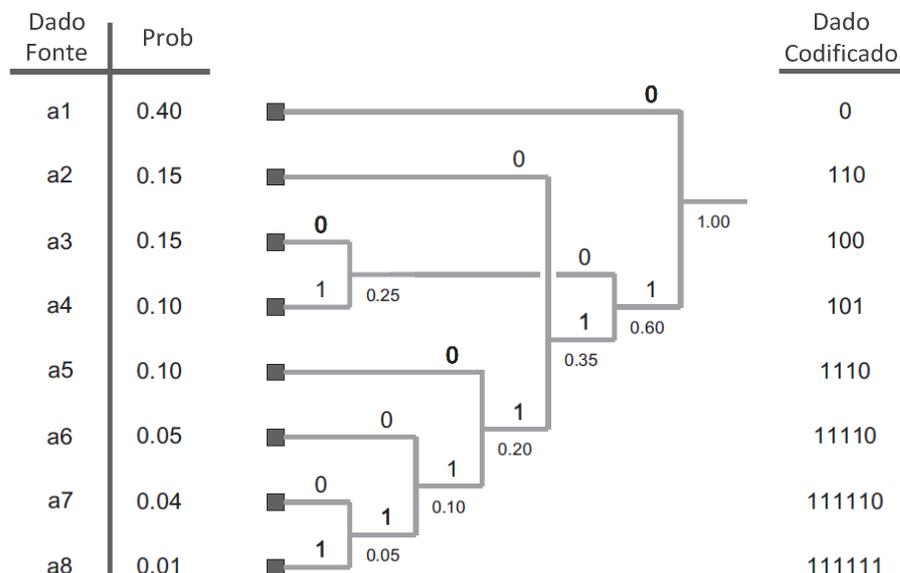


Figura 20 – Exemplo de um processo de construção dos Códigos de Huffman

Fonte: (DUTOIT e MARQUÉS, 2009) – Tradução do próprio autor

Através da Figura 20, verifica-se o potencial de um codificador de entropia. O dado de entrada *a1*, que tem maior probabilidade de ocorrer, é representado apenas com um bit. Já o dado *a8* é representado por seis bits, devido sua probabilidade de ocorrência ser baixa.

Desta forma, o codificador MPEG-4 AAC conta com 12 tabelas dos códigos de Huffman para os coeficientes espectrais, e uma tabela com os Códigos de Huffman para os valores dos *scalefactors* (ISO/IEC, 2004). Durante o processo de codificação, uma das 12 tabelas dos códigos de Huffman deve ser escolhida para melhor representar os coeficientes espectrais em cada *scalefactor band*. Na Tabela 3 e Tabela 4 são apresentados os parâmetros dos Códigos de Huffman para os *scalefactors* e para os coeficientes espectrais.

Número do Código	Dimensão do Código	Faixa de Valores
0	1	-60 a 60

Tabela 3 – Parâmetros dos códigos de Huffman para *scalefactors*

Fonte: (ISO/IEC, 2004) – Tradução do próprio autor

Número do Código	Sem sinal (<i>unsigned</i>)	Dimensão do Código	LAV para o Código
0	-	-	0
1	0	4	1
2	0	4	1
3	1	4	2
4	1	4	2
5	0	2	4
6	0	2	4
7	1	2	7
8	1	2	7
9	1	2	12
10	1	2	12
11	1	2	(16) ESC

Tabela 4 – Parâmetros dos códigos de Huffman para o espectro de frequências

Fonte: (ISO/IEC, 2004) – Tradução do próprio autor

Na Tabela 3 e 4, a 1ª coluna determina o número do código, ou a quantidade de códigos disponíveis no codificador para cada tipo de dado (*scalefactors* ou componentes espectrais). A dimensão do código representa a quantidade de dados da fonte que serão utilizados na codificação. No caso de *scalefactors*, os dados são utilizados sozinhos, sem qualquer agrupamento. Porém, para os códigos de Huffman, os dados são agrupados em duas-duplas e quatro-duplas, de acordo com número do código (ISO/IEC, 2004).

Os valores de *scalefactors* que podem ser codificados pelo código de Huffman, são valores compreendidos entre -60 e 60. Para a representação dos coeficientes espectrais, este limite é definido pela 4ª coluna da Tabela 4. Assim *LAV* (*Largest Absolute Value*) significa o maior valor absoluto permitido pelo código de Huffman dos coeficientes espectrais quantizados. O caso especial (*ESC – Special*

Codebook) é para o código 11, que permite codificar coeficientes quantizados com valores maiores do que 16 (ISO/IEC, 2004). Alguns códigos de Huffman permitem a utilização de valores com sinal e outros sem sinal, este parâmetro é definido na 2ª coluna da Tabela 4. O parâmetro para valores sem sinal é definido como 1, e para com sinal definido como 0 (ISO/IEC, 2004).

3.2.8 Predição

Predição é uma técnica usada pelo codificador, que procura prever o sinal de áudio a partir de sua dinâmica atual. Isso permite uma melhora considerável em seu desempenho em relação à redução de redundância temporal de um sinal de áudio (ISO/IEC, 2004). Assim este processo explora a correlação entre as componentes espectrais de *frames* consecutivos. Devido os *frames short blocks* em sua própria definição, já serem definidos por características não estacionárias de sinal, a predição é somente utilizada em *frames long blocks*, *long start blocks* e *long end blocks* (ISO/IEC, 2005).

Os valores preditos são calculados para cada *scalefactor band* de cada *frame long block*, a partir dos valores dos coeficientes espectrais atuais. Desta forma, é calculado o erro residual entre as componentes espectrais estimadas e as componentes reais conforme a equação (21) (ISO/IEC, 2004):

$$e(n) = x(n) - x_{est}(n) \quad (21)$$

onde:

$e(n)$ = erro residual;

$x(n)$ = componente espectral atual;

$x_{est}(n)$ = componente espectral estimada;

Depois de calculado o erro residual, o codificador avalia se é melhor quantizar e transmitir apenas o erro residual, ou o coeficiente espectral, verificando qual das duas situações necessitam de uma quantidade menor de bits em sua representação (ISO/IEC, 2004). Deve-se observar que ao optar pela transmissão do erro residual, o codificador deve informar esta situação ao decodificador. Isso implica, em transmitir

junto com o erro residual, informações auxiliares, que incrementam a quantidade de bits necessários na taxa total do codificador. Para momentos do sinal de áudio estacionários, a etapa de predição contribui efetivamente na redução da quantidade de bits (WATKINSON, 2001).

3.2.9 Perfis e Níveis do MPEG-4 AAC

O codificador MPEG-4 AAC tem como algumas de suas principais características suportar até 48 canais de áudio, permitindo os formatos padrões: mono (1 canal), estéreo (2 canais) e 5.1 (6 canais). A máxima frequência de amostragem possível em que o codificador pode operar é 96kHz.

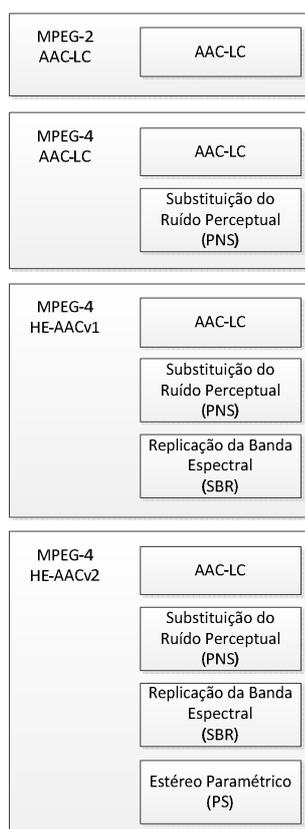


Figura 21 – Perfis do codificador de áudio AAC

Fonte: (MELTZER e MOSER, 2006) – Tradução do próprio autor

Com o objetivo de permitir uma estrutura modular, com diferentes ferramentas e estratégias incorporadas ao codificador, foram criados perfis e níveis que englobam diferentes características de codificação. Diante de diversas alterações e

inovações, o codificador AAC é dividido em quatro perfis: MPEG-2 AAC *Low Complexity (LC)*, MPEG-4 AAC *Low Complexity (LC)*, MPEG-4 *High Efficiency AAC version 1* e MPEG-4 *High Efficiency AAC version 2*. Na Figura 21 é possível visualizar os perfis do codificador AAC.

Para o sistema brasileiro de televisão digital, a norma ABNT restringiu alguns perfis do codificador AAC, determinando quais deveriam ser implementados pelos dispositivos aderentes à norma. Este trabalho abordará o codificador MPEG-4 AAC com as restrições da norma brasileira de TV Digital. Desta forma, os seguintes perfis e níveis do padrão MPEG-4 AAC fazem parte da norma brasileira de televisão digital (ABNT, 2007):

a) *LC (Low Complexity)*, perfil básico do padrão AAC; níveis L2 e L4;

b) *HE (High Efficiency)*, perfil avançado de alta eficiência, combinando o perfil *LC* com o uso da ferramenta *SBR (Spectral Band Replication)* para a versão 1 deste perfil, níveis L2 e L4;

c) *HE* combinado à ferramenta *PS (Parametric Stereo)* para a versão 2 deste perfil; nível L2.

Na Tabela 5 é possível identificar as diferentes características dos níveis para o perfil HE-AAC. A norma apresenta as unidades *PCU (Processor Complexity Units)* e *RCU (RAM Complexity Units)* que medem a complexidade de implementação de determinado nível do codificador. Equipamentos que implementam o codificador AAC, devem ter estas especificações de hardware atendidas.

Nível	Máximo Número de canais	Max. Frequência de Amostragem (kHz) SBR-não presente	Max. Frequência de Amostragem (kHz) SBR-presente	Unidade de Complexidade do Processador (PCU)	Unidade de Complexidade da RAM (RCU)
1	-	-	-	-	-
2	2	48	24	9	10
3	2	48	48	15	10
4	5	48	24/48	25	28
5	5	96	48	49	28

Tabela 5 – Níveis para o perfil HE-AAC

Fonte: (ISO/IEC, 2005) – Tradução do próprio autor

3.3 Qualidade Perceptual do Áudio

Uma questão interessante que surge ao se estudar codificadores de áudio perceptuais é de como avaliar o desempenho dos mesmos de uma forma objetiva. A experiência auditiva de cada ouvinte é algo muito subjetivo, o que é bom para uma pessoa, pode não ser para outra. Desta forma, a maneira mais adequada para poder medir a qualidade de áudio de maneira objetiva, é utilizar uma métrica que possua o modelo do ouvido humano em sua concepção (FASTL e ZWICKER, 2007). Nos mais modernos e recentes sistemas de codificação de áudio, o *NMR* (*Noise-to-Mask Ratio*) é utilizado como uma medida da qualidade áudio perceptual (NIAMUT, 2006), e assim permite avaliar a qualidade de um codificador de áudio. Como é uma relação entre dois níveis de sinal, o nível do ruído sobre o nível da máscara, a medida *NMR* pode ser expressa em dB (POHLMAN, 1995).

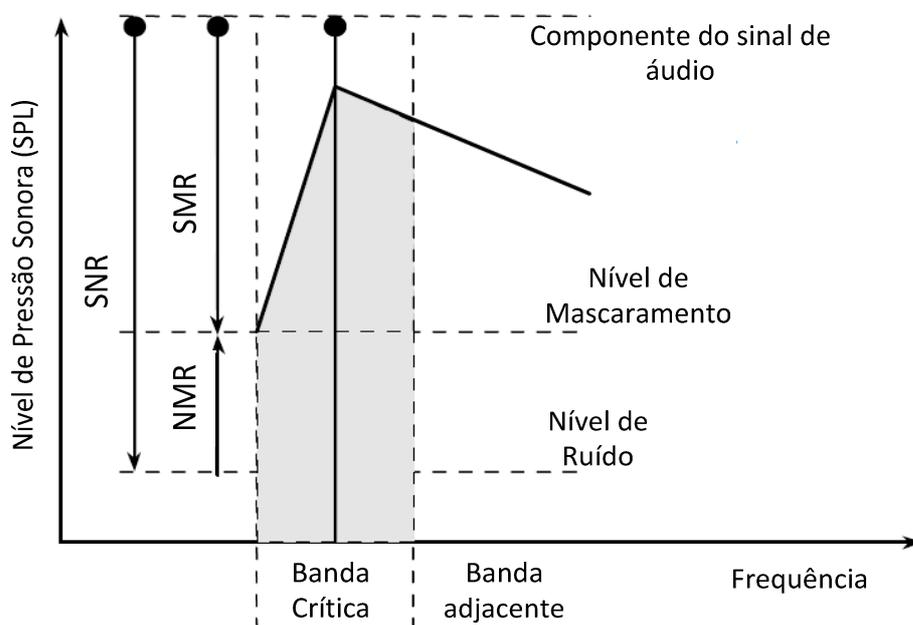


Figura 22 – Medida de qualidade objetiva através do *NMR*

Fonte: (PAINTER e SPAN, 2000) – Tradução do próprio autor

A Figura 22 apresenta uma componente de áudio, situada dentro de uma determinada banda crítica. Esta componente gera uma máscara, de maneira que todo ruído que estiver abaixo desta, não será percebido pelo ouvido humano. Desta forma, a medida *NMR* expressa a quantidade de ruído existente dentro de uma

banda crítica em relação à máscara do ouvido (Figura 8). Assim o *NMR* é calculado para cada *scalefactor band* do *frame*, fazendo a relação entre o limite de audição da máscara e o erro de quantização inserido pelo codificador (WATKINSON, 2001). Ao final do cálculo do *NMR* de todos os *scalefactor bands*, a qualidade do *frame* pode ser medida através de um *NMR* médio, ou seja, *ANMR*. Na verdade existem duas medidas que derivam do *NMR*, o *MNMR* (*Maximum Noise-to-Mask Ratio*) que define a máxima relação entre o ruído e o nível da máscara, e o *ANMR* (*Average Noise-to-Mask Ratio*) que é a relação média, contida em um *frame*, entre os sinais de ruído e máscara (BAUER, 2005).

Quanto menor o valor da relação *NMR*, mais abaixo estará o ruído gerado pelo processo de quantização do limiar audível do ouvido humano. Nesta situação, menos perceptível será o ruído sobre o sinal de áudio. Porém, para maiores valores de *NMR*, mais significativa será a distorção sobre o sinal de áudio, e assim esta informação pode ser utilizada pelo codificador, para tentar eliminar esta distorção através dos laços de iterações interno e externo (POHLMAN, 1995).

Na Figura 22, também é possível verificar junto com a medida *NMR*, outras duas medidas, o *SNR* (*Signal-to-Noise Ratio*) e *SMR* (*Signal-to-Mask Ratio*). A primeira medida descreve a relação entre o nível da componente do sinal de áudio e o nível de ruído. Quanto maior for esta relação, menos o sinal é interferido pelo ruído. A segunda relação *SMR*, de maneira análoga ao *SNR*, também descreve a relação entre dois níveis, o nível da componente do sinal de áudio e o nível de mascaramento. Esta relação descreve o quanto o nível da componente do sinal de áudio está acima do nível de mascaramento. O *NMR* pode ser obtido através destas duas relações, fazendo a subtração entre *SNR* e *SMR* conforme a equação (22) (PAINTER e SPAN, 2000). Através de uma análise gráfica da Figura 22, também é possível identificar esta operação de subtração.

$$NMR = SNR - SMR \text{ (dB)} \quad (22)$$

Devido à medida *NMR* ser a mais amplamente utilizada para avaliação objetiva da qualidade de um sinal de áudio (BAUER e VINTON, 2006), este trabalho utilizou esta métrica como principal ponto de avaliação durante o desenvolvimento e validação da estrutura de otimização do codificador MPEG-4 AAC. Como a *NMR* é

uma medida da qualidade perceptual do áudio, a Função Objetivo do problema de codificação AAC, possui esta métrica inserida em sua concepção.

4 METAHEURÍSTICAS

Em 1986, Glover (GLOVER, 1986) em seu artigo utilizou pela primeira vez o os termos metaheurística e Busca Tabu, que hoje são amplamente utilizados na literatura (GLOVER e LAGUNA, 1997). Metaheurística é um processo de geração iterativa que orienta uma heurística subordinada que combina, de forma inteligente, estratégias diferentes para investigar o espaço de soluções. Assim, com uma boa diversificação do processo de busca através do espaço de soluções, a metaheurística se aproxima de regiões promissoras, encontrando boas soluções para o problema desejado (REEVES, 1993).

É interessante observar que as metaheurísticas podem ser utilizadas para solucionar problemas de diversas naturezas, como por exemplo problemas de escalonamento (*Scheduling*) (GAREY, JOHNSON e SETHI, 1976) (MURATA e ISHIBUCHI, 1994), problema de Seleção de Partes (*Part Selection*) (SRIVASTAVA e CHEN, 1993) (JEON e LEEP, 2005), problema de roteamento de veículos (*Routing*) (GENDREAU, HERTZ e LAPORTE, 1994) (GENDREAU, LAPORTE e POTVIN, 2001), entre outros (GLOVER e LAGUNA, 1997). O problema de otimização que será abordado neste trabalho é o Problema de Codificação AAC (BAUER, FELLERS e DAVIDSON, 2006) que será detalhado no capítulo 6.

Algumas das principais metaheurísticas encontradas em aplicações na literatura são: Busca Tabu (*Tabu Search*), Algoritmos Genéticos (*Genetic Algorithms*), Têmpera Simulada (*Simulated Annealing*) e GRASP (*Greedy Randomized Adaptive Procedure*) (VIANA, 1998). Neste trabalho será utilizada uma estrutura híbrida utilizando as metaheurísticas Algoritmo Genético e Busca Tabu.

4.1 Busca Tabu

Proposta por Glover (1986), a Busca Tabu é um método de busca em trajetória que se baseia na melhoria da solução inicial, através da busca de novas soluções pela geração de vizinhanças. Fazendo uso de uma Lista Tabu de movimentos, a Busca Tabu pode escapar de mínimos locais e platôs e explorar o espaço global de soluções (GLOVER e LAGUNA, 1997).

De forma básica, a Busca Tabu possui três princípios fundamentais (VIANA, 1998):

- Uso de uma estrutura de dados (fila) para armazenar o histórico do processo de busca;
- Uso de uma ferramenta de controle para um balanceamento entre a aceitação, ou não, de uma nova configuração, com base nas informações registradas na Lista Tabu, referente às restrições e à Função Objetivo (FO);
- Incorporação de procedimentos que alternam as estratégias de diversificação e intensificação, como é ilustrado na Figura 23.

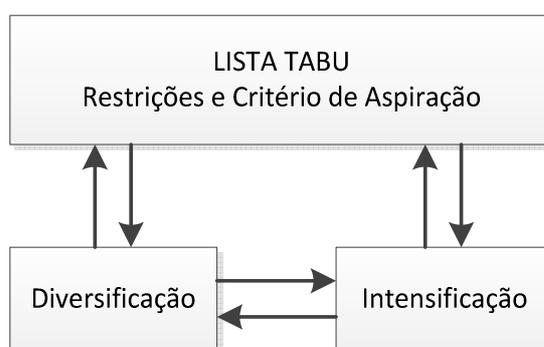


Figura 23 – Estratégia utilizada pela Busca Tabu

Fonte: (VIANA, 1998)

Para a técnica da Busca Tabu, ou qualquer outra metaheurística, é fundamental a definição da função objetivo do problema que se deseja solucionar. Assim, esta é definida matematicamente junto às restrições, formando o modelo matemático do problema. Efetivamente, dentro do laço de iterações das metaheurísticas, é a função objetivo que avalia o quão boa ou ruim é uma determinada solução.

Para iniciar a trajetória gerada pelo processo de busca, é necessária a criação de uma solução inicial. Esta pode ser gerada aleatoriamente ou utilizando um processo baseado em uma heurística de busca local, de maneira que esta solução necessariamente seja viável, ou seja, esteja localizada dentro do espaço de soluções factíveis do problema (GLOVER, 1989).

Assim, a partir de uma solução inicial “s”, são geradas novas soluções vizinhas utilizando os métodos de estruturas de vizinhança. Desta forma é gerado um subconjunto V de $N(s)$ e realizado o movimento para a melhor solução s^* em V . A utilização do critério de melhor movimento na Busca Tabu é baseada na suposição de que movimentos bem avaliados têm grande probabilidade de conduzir a uma ótima solução ou ao menos próxima dela (GLOVER, 1989).

A fim de evitar ciclos e mínimos locais, o algoritmo não permite voltar às soluções que já foram previamente visitadas em um número “ k ” de movimentos anteriores. Esta estratégia é denominada de Lista Tabu (GLOVER e LAGUNA, 1997). Como a lista possui um tamanho finito “ k ”, sempre que um determinado movimento é executado, por exemplo de “s” para s^* , o movimento é introduzido no fim da lista e o movimento mais antigo é removido. A partir deste momento, todos os movimentos que retornam a “s” ficam proibidos nas próximas “ k ” iterações. Desta forma, “s” tornou-se tabu, e qualquer movimento que leve a “s” é considerado não permitido (VIANA, 1998).

É essencial que a Lista Tabu possua um comprimento definido e finito, pois poderá haver momentos em que será necessário utilizar um movimento passado para poder conduzir a trajetória da busca a boas soluções (GLOVER e LAGUNA, 1997). O tamanho da vizinhança também é outro parâmetro importante para o algoritmo Busca Tabu. Uma vizinhança pequena ou grande pode comprometer negativamente a execução do algoritmo. Assim, este valor deve ser ajustado juntamente com o número máximo de iterações e o tamanho da Lista Tabu (VIANA, 1998).

Proibir um número muito grande de movimentos, também poderá afetar o algoritmo, fazendo o mesmo perder desempenho, devido aos ciclos repetidos de soluções. Assim, é importante que um determinado movimento, após um número de iterações, venha a sair da Lista Tabu (GLOVER, 1989).

Existem movimentos que são considerados Tabu, porém podem levar a solução a uma melhoria e, desta forma, é introduzido um procedimento para cancelar a situação de Tabu de um determinado movimento. Este critério de retirada do movimento é chamado de Critério de Aspiração (VIANA, 1998). Assim, uma função de critério de aspiração $A(z)$ é definida para cada valor “z” da função objetivo.

Se o movimento que leva para a solução vizinha s^* está na Lista Tabu, porém $F(s^*) < A(z=F(s))$, então s^* é considerada como uma solução normal de V , e dessa forma o movimento pode ser executado (GLOVER e LAGUNA, 1997) (VIANA, 1998).

O critério de parada do algoritmo de Busca Tabu pode ser definido por duas regras. Uma delas é pela definição de um número máximo de iterações (conhecido na literatura como BT_Max), que serão realizadas sem obter uma solução melhor, do que a melhor solução já obtida. A outra regra condiciona o critério de parada, quando a melhor solução encontrada é considerada muito próxima da solução ótima (GLOVER e LAGUNA, 1997). Porém, para que este procedimento possa ser utilizado é necessário conhecer a solução ótima. A Figura 24 apresenta o pseudocódigo para a metaheurística Busca Tabu.

```

Seja  $s_0$  solução inicial;
 $s^* \leftarrow s$ ; {Melhor solução obtida até então}
Iter  $\leftarrow$  0; {Contador do número de iterações}
Melhor_Iter  $\leftarrow$  0; {Iteração mais recente que forneceu  $s^*$ }
Seja BT_max o número máximo de iterações sem melhora em  $s^*$ ;
 $T \leftarrow \emptyset$ ; {Lista Tabu}
Inicialize a função de aspiração A;

ENQUANTO (Iter – Melhor_Iter  $\leq$  BT_max) FAÇA
    Iter  $\leftarrow$  Iter + 1;
    Seja  $s' \leftarrow s \oplus m$  o melhor elemento de  $V \subseteq N(s)$  tal que o movimento  $m$  não seja tabu ( $m \notin T$ )
    ou  $s'$  atenda à condição de aspiração ( $F(s') < A(F(s))$ );
    Atualize a Lista Tabu T;
     $s \leftarrow s'$ ;

    SE  $F(s) < F(s^*)$  ENTÃO
         $s^* \leftarrow s$ ;
        Melhor_Iter  $\leftarrow$  Iter ;
    FIM-SE;

    Atualize a função de aspiração A (caso  $s'$  atenda à condição de aspiração);
FIM-ENQUANTO;
Retorne  $s^*$ ;
fim BT;
```

Figura 24 – Pseudocódigo do algoritmo Busca Tabu para um problema de minimização

Fonte: (VIANA, 1998)

4.2 Algoritmo Genético

Em 1975 em seu livro *Adaptation in Natural and Artificial Systems*, Holland apresentou os Algoritmos Genéticos, fundamentando toda teoria da metaheurística como uma abstração dos conceitos da evolução biológica (HOLLAND, 1975). Assim, os Algoritmos Genéticos (AG) pertencem a uma família de métodos heurísticos de busca aleatória embasados na analogia da evolução biológica. AGs são algoritmos de busca baseados nos mecanismos de seleção natural e genética natural. Eles combinam a sobrevivência de indivíduos mais aptos, com uma troca de informações aleatória entre indivíduos da população (GOLDBERG, 1989). Os componentes básicos de um AG são a população de indivíduos e os operadores aleatórios que são introduzidos no modelo de mutação e cruzamento. Um determinado indivíduo corresponde a uma tentativa de solução. Cada um é codificado por uma sequência de tamanho fixo, denominada cromossomo, composto por símbolos, chamados genes (VIANA, 1998).

De acordo com a programação do Algoritmo Genético, o mesmo pode ser utilizado para resolver problemas de diferentes tipos, como por exemplo o problema de Seleção de Partes (JEON e LEEP, 2005) e o Problema de *Scheduling* (MURATA e ISHIBUCHI, 1994). Porém, para diferentes problemas, existem diferentes maneiras de modelar a função objetivo, as restrições e a codificação utilizada no cromossomo.

4.2.1 Codificação do Cromossomo

Como cada cromossomo representa uma solução para o problema que se deseja solucionar utilizando um AG, o processo de codificação determina como as informações da solução estarão organizadas no cromossomo. Para cada tipo de problema, os genes do cromossomo representam diferentes variáveis de decisão da FO.

Uma maneira de representar um cromossomo de um determinado problema é fazendo o uso de vetores. Assim, cada elemento do vetor passa a representar uma variável de decisão do problema. É interessante observar que cada gene pode assumir valores inteiros, binários, reais ou uma combinação destes.

4.2.2 Função Objetivo

O Algoritmo Genético consiste em gerar a cada iteração, uma nova geração contendo novos indivíduos com base nas gerações passadas, esperando uma evolução da mesma. Para que cada nova geração de indivíduos possa ser avaliada, de maneira a medir o quão boas ou ruins são as soluções de uma determinada geração, é utilizada a Função Objetivo. Também conhecida como *fitness*, esta permite o cálculo da aptidão de cada indivíduo da população para o problema em análise. Caso seja um problema de maximização, indivíduos com maior valor para a FO têm maior probabilidade de se manterem dentro da população e serem escolhidos para gerarem novos indivíduos de futuras gerações. De maneira análoga para um de problema de minimização, os melhores indivíduos são aqueles que possuem um menor valor para a FO (POLTOSI, 2007).

A função objetivo é construída durante o processo de modelagem do problema a ser analisado e solucionado. Desta forma, é fundamental que a função objetivo retrate de maneira fidedigna o problema que espera-se que a metaheurística solucione.

4.2.3 Operadores de Seleção

Para que se possam determinar quais serão os indivíduos de uma determinada geração que farão parte da geração sucessora, ou quais indivíduos que serão selecionados como pais de novos indivíduos, são utilizados os operadores de seleção. Através destes, os indivíduos são selecionados da população atual, sendo que os mais aptos tem uma maior probabilidade de serem escolhidos. Existem diferentes técnicas para selecionar indivíduos como, por exemplo, Seleção por Torneio, Seleção Elitista, Seleção por Roleta, entre outros (JADAAN, RAJAMANI e RAO, 2008).

4.2.4 Operadores de Cruzamento e Mutação

A população inicial de indivíduos pode ser criada de maneira totalmente aleatória, ou ser híbrida, combinando uma criação aleatória com alguma heurística

local. Para obter novas gerações que possuam características diferentes, mas que também carreguem elementos herdados da geração anterior, são utilizados os operadores de cruzamento e mutação (GOLDBERG, 1989).

O processo de cruzamento de indivíduos consiste em, aleatoriamente ou não, escolher dois indivíduos pais para que gerem dois novos indivíduos filhos. Estes dois novos indivíduos serão duas novas soluções para o problema proposto, que possuem genes provenientes de ambos os pais. Espera-se que estas soluções sejam melhores que as já obtidas. Na Figura 25 é apresentado o cruzamento de ponto único de corte.

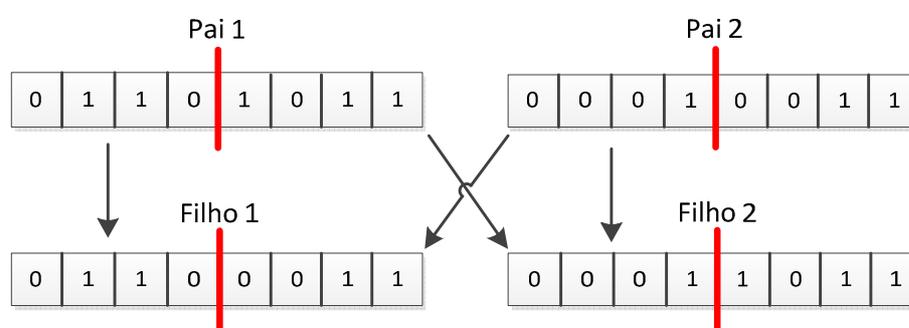


Figura 25 – Operador de cruzamento de ponto único do AG

Neste método de cruzamento, é sorteado um ponto do cromossomo dos indivíduos, que define um ponto de corte. Este ponto divide ambos os pais em duas partes. Os dois novos filhos, são gerados combinando a primeira parte do Pai 1 com a segunda parte do Pai 2, e combinando a primeira parte do Pai 2 com a segunda parte do Pai 1. Outro método de recombinação genética é o cruzamento aritmético apresentado na Figura 26.

Para o cruzamento aritmético apresentado na Figura 26, é gerado apenas um filho. Neste processo é realizada uma média aritmética simples, entre os valores contidos em cada gene do cromossomo. Desta forma, o resultado da média calculada, define os genes do cromossomo do novo indivíduo criado.

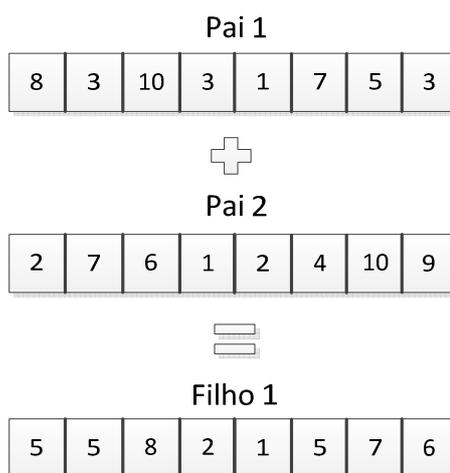


Figura 26 – Operador de cruzamento aritmético do AG

Para a população receber novos indivíduos, é conveniente que os “menos promissores” sejam eliminados, dando espaço para outros com características diferentes.

Para que nesta etapa de criação de novos indivíduos haja certa diversificação, aleatoriamente podem ser sorteados indivíduos para sofrerem mutação (Figura 27). Este processo consiste em alterar, também de forma aleatória, alguns genes do indivíduo, atendendo às regras de mutação determinadas pelo projetista do algoritmo.



Figura 27 – Operador de mutação do AG

O processo de iteração que cria novas gerações se repete até que o critério de parada seja satisfeito. O critério de parada pode ser definido como sendo um número máximo de gerações, ou um determinado tempo de processamento. Na Figura 28 é apresentado o pseudocódigo da metaheurística Algoritmo Genético.

```
Gerar população inicial;  
Avaliar cada indivíduo da população;  
ENQUANTO critério de parada não for satisfeito FAÇA  
{  
    Selecionar indivíduos pais;  
    Criar novos indivíduos através do cruzamento e mutação;  
    Armazenar novos indivíduos na nova população;  
    Avaliar cada indivíduo da nova população;  
}  
Apresentação do melhor indivíduo;
```

Figura 28 – Pseudocódigo do Algoritmo Genético

Fonte: (VIANA, 1998)

Durante o processo de implementação do AG é necessária a configuração de alguns parâmetros que definem a maneira de operação do algoritmo. São eles: tamanho da população, probabilidade de cruzamento, probabilidade de mutação e critério de parada. Desta forma, com o algoritmo configurado, espera-se que as novas gerações sejam oportunidades para que o algoritmo possa percorrer o espaço de soluções, e procurar convergir para soluções de boa qualidade.

5 TRABALHOS RELACIONADOS

Na literatura é possível encontrar alguns trabalhos que abordam o Problema de Codificação AAC, utilizando diferentes técnicas de solução e algumas variações na estrutura do modelo utilizado. Estas variações são no sentido de simplificar o modelo, de maneira que se possam utilizar métodos de solução mais simples para resolvê-lo, e também no sentido de enriquecer o modelo com o acréscimo de variáveis de decisão, de modo a deixá-lo mais fiel à realidade, e produzir resultados com qualidade perceptual de áudio superior. O problema deste último é que com o acréscimo de detalhes, o problema se torna mais complexo de se resolver, e assim os métodos matemáticos de solução necessitam de um tempo computacional superior para sua resolução.

No trabalho de Aggarwal, Regunathan e Rose (2000), um algoritmo denominado *Trellis Search*, baseado em um arranjo de treliça é apresentado para melhorar a eficiência do codificador de áudio AAC. A proposta do trabalho é que a etapa de quantização do codificador seja mais eficiente, de maneira que possa produzir na saída taxas de bits menores quando comparado ao codificador de referência. Nesta modelagem, para simplificar o problema, os autores utilizaram apenas duas variáveis de decisão, os *scalefactors* e os códigos de Huffman. O modelo assume como referência na função objetivo, a quantidade de bits necessários para representar os coeficientes espectrais quantizados, e também as informações auxiliares, preservando a qualidade perceptual do áudio, ou seja, minimizando a métrica *ANMR/MNMR*. O problema proposto foi resolvido utilizando a Busca *Viterbi*, método que consiste em encontrar a combinação das variáveis de decisão que determinam o menor custo dentro da treliça. No trabalho são assumidos 60 *scalefactors* e 12 códigos de Huffman, de forma que o número total de todas as possíveis combinações de duas *scalefactor bands* é $60 \times 12 \times 2 = 1440$. Desta forma, para reduzir a complexidade do problema, os autores restringiram as possibilidades de escolha dos códigos de Huffman, para no máximo quatro códigos vizinhos da escolha atual. Os resultados obtidos pelo trabalho demonstram uma melhoria na qualidade perceptual de áudio, em relação ao codificador de referência, para uma mesma taxa de bits. A melhoria alcançada pelo trabalho resultou em um incremento de aproximadamente duas vezes na qualidade perceptual de áudio, o que representa um decréscimo de duas vezes na relação *ANMR*.

Dando sequência ao trabalho de Aggarwal, os autores Bauer e Vinton (2006), propuseram uma melhoria na forma de solução do problema de codificação AAC. Ainda utilizando apenas duas variáveis de decisão, os *scalefactors* e os códigos de Huffman, os autores propuseram uma técnica diferente para resolver o Problema de Codificação AAC. Denominada pelos autores como *Fast Trellis Search*, a técnica baseia-se no processo da Busca *Viterbi* utilizada por Aggarwal, porém com algumas modificações. O Método dos Multiplicadores de Lagrange, utilizado na Busca *Viterbi*, consiste em converter problemas de minimização com restrições, em problemas sem restrições, inserindo um novo parâmetro: o Multiplicador de Lagrange (λ). Na técnica *Fast Trellis Search* o Método dos Multiplicadores de Lagrange é utilizado como função da Entropia Perceptual, e assim a solução de menor custo da treliça pode ser obtida sem percorrer todos os possíveis multiplicadores de Lagrange, diminuindo a complexidade do problema. A função objetivo baseia-se na métrica *ANMR*, de maneira que sua minimização conduz à solução do problema de codificação, para baixas taxas de bits mantendo a qualidade perceptual do áudio. Para análise dos resultados, foram utilizados como entrada arquivos *WAV* com apenas um canal (mono), amostrados a uma frequência de 44100Hz. O método *Fast Trellis Search*, obtém uma qualidade perceptual muito próxima ao do algoritmo *Trellis Search* abordado por Aggarwal, porém resolve o problema de codificação aproximadamente 25 vezes mais rápido. Devido ao tempo computacional necessário para a resolução do problema, a técnica é incapaz de viabilizar uma implementação do codificador AAC em tempo real.

Nos artigos “*Multidimensional Optimization of MPEG-4 AAC Encoding*” (BAUER, FELLERS e DAVIDSON, 2006) e “*The Optimal Choice of Encoding Parameters for MPEG-4 AAC Streamed over Wireless Networks*” (BAUER, 2005), os autores também abordam o Problema de Codificação AAC. Porém, nestes trabalhos o modelo não utiliza apenas duas variáveis de decisão, e assim torna o problema mais fiel à realidade e ao mesmo tempo mais complexo. Utilizando Programação Linear Inteira Mista (*MILP*) o problema é modelado fazendo uso de quatro variáveis de decisão: os blocos dos frames, o agrupamento dos blocos, os *scalefactors* e os Códigos de Huffman. O modelo matemático que descreve o problema para quatro variáveis de decisão é complexo e extenso, e possui cerca de quarenta equações de restrições. Na realização dos testes foram utilizados arquivos *WAV* de apenas um

canal (mono) com duração de aproximadamente 100s e frequência de amostragem de 44100Hz. Nesta abordagem, os autores conseguem obter melhoras significativas na qualidade perceptual do áudio, quando comparados aos modelos das estruturas dos codificadores antes estudados e de referência. Porém, devido à complexidade do problema, ao manipular quatro variáveis de decisão, o tempo de processamento acaba sendo muito elevado. Isso acaba não permitindo que o codificador com esta estrutura de otimização dos parâmetros, seja implementado em tempo real. Com intuito de reduzir o tempo de execução, os autores relaxaram algumas restrições da formulação *MILP*. Desta forma, foi possível melhorar o desempenho do codificador quanto ao tempo de execução, porém houve uma perda na qualidade perceptual. Apesar desta prática, o ganho em desempenho não foi significativo suficientemente para implementações em tempo real.

No trabalho (ALI e SALEEM, 2008), os autores propõem uma técnica que impacta na melhoria da qualidade perceptual do áudio, especificamente para uma taxa fixa de 48kbps. A técnica proposta baseia-se em um novo método de otimização do processo de quantização, substituindo o tradicional método *TLS* dos codificadores MPEG-4 AAC. Ao contrário do codificador de referência que utiliza um conjunto fixo de divisão de bandas, os autores propõem que as bandas do codificador sejam dinamicamente selecionadas em função do sinal de áudio e do modelo psicoacústico. Através deste novo elemento inserido, puderam identificar uma melhora no desempenho do laço de iteração *TLS*. O desenvolvimento do modelo matemático utilizado na estrutura de otimização proposta, baseia-se no trabalho de Bauer (2005). Assim, através da realização de testes subjetivos com ouvintes e também através da medida de qualidade perceptual do áudio, identificaram melhorias do codificador proposto quando comparado ao codificador de referência, porém apenas para baixas taxas como 48kbps.

Na Tabela 6 é possível visualizar as diferentes características entre alguns dos trabalhos relacionados.

Trabalho	Variáveis de decisão	Método de solução	Resultados quando comparado ao codificador MPEG-4 AAC de referência	Complexidade do Algoritmo
Aggarwal, Regunathan e Rose (2000)	<i>ScaleFactors</i> e Códigos de Huffman	Busca <i>Viterbi</i> (<i>Trellis Search</i>)	Obteve melhorias, com diminuição pela metade do valor da distorção.	Incremento na complexidade do algoritmo do codificador. Inviável implementação em tempo real.
Bauer e Vinton (2006)	<i>ScaleFactors</i> e Códigos de Huffman	<i>Fast Trellis Search</i>	Aproximadamente mesma eficiência do que a <i>Trellis Search</i> .	25 vezes mais rápido que a <i>Trellis Search</i> . Inviável implementação em tempo real.
Bauer, Fellers e Davidson (2006) e Bauer (2005)	Blocos dos <i>frames</i> , Agrupamento dos blocos, <i>ScaleFactors</i> , Códigos de Huffman	Programação Linear Inteira Mista com algumas restrições relaxadas.	Significativa melhora em relação às soluções existentes.	Necessário altíssimo tempo computacional. Inviável implementação em tempo real.
Ali e Saleem (2008)	<i>ScaleFactors</i> e Códigos de Huffman	Técnica de determinação dinâmica das bandas no processo de quantização.	Melhorias na qualidade perceptual apenas para a taxa de 48kbps.	Menos complexo em relação ao método <i>TLS</i> .

Tabela 6 – Comparativo trabalhos relacionados

A abordagem dos autores Lan-Ying e Min (2008), enfatiza o uso das metaheurísticas Busca Tabu e Algoritmo Genético em uma solução híbrida. O problema relatado no trabalho é um problema clássico de sistemas embarcados. O problema aborda um dos passos mais importantes no desenvolvimento de um projeto deste tipo, que é a decisão de quais componentes do sistema devem ser implementados em *software* e quais em *hardware*. Desta maneira, os autores propuseram um modelo computacional utilizando uma solução híbrida de metaheurísticas, para resolver o problema de Seleção de *hardware-software* em sistemas embarcados. Na modelagem foram utilizadas quatro variáveis de decisão: custo do *software*, tempo de *software*, custo do *hardware* e tempo de *hardware*. Os experimentos foram realizados não somente utilizando a solução combinada das metaheurísticas, como também das metaheurísticas separadas. Os resultados apresentados demonstram que a solução combinada da Busca Tabu e Algoritmo Genético, produz melhores soluções para a FO, do que as soluções obtidas com as duas metaheurísticas operando separadas.

Na literatura pôde-se encontrar também uma abordagem de uma solução híbrida de metaheurísticas, utilizando Busca Tabu e Algoritmo Genético, aplicada a um modelo de otimização do codificador de vídeo H.264/AVC utilizado no Sistema Brasileiro de Televisão Digital. Esta estrutura foi denominada de Simulador de

Metaheurísticas aplicadas ao CODEC. No trabalho (GÓMEZ e LINCK, 2012), os autores apresentam um modelo utilizando seis variáveis de decisão: taxa de bits, taxa de frames, quantização do quadro I, quantização do quadro B, quantização do quadro P e o número de quadros B. O modelo baseia-se em um somatório ponderado das variáveis de decisão, de maneira que cada variável de decisão possui um determinado peso. Estes pesos foram determinados através de exaustivos experimentos, confrontando o comportamento dinâmico das variáveis do modelo, com os dados encontrados na literatura. É interessante observar nesta abordagem, que a estrutura de otimização dos parâmetros do codificador de vídeo, não é executada em tempo real. Uma vez encontrados os parâmetros, estes são então carregados no codificador H.264/AVC para que ele execute o processo de codificação de vídeo, configurado com os parâmetros definidos pela solução híbrida de metaheurísticas. Através deste trabalho, os autores identificaram que o modelo proposto representa adequadamente a dinâmica das variáveis de codificação. Verificaram também que o Simulador de Metaheurísticas, aplicado ao CODEC, procura melhorar a configuração do codificador H.264, através da melhor combinação dos seis parâmetros apresentados.

6 MODELO DE SIMULAÇÃO DO CODIFICADOR MPEG-4 AAC

O principal objetivo em codificar um sinal de áudio é reduzir a quantidade de bits necessários para representá-lo, mantendo a qualidade perceptual. Assim, o codificador MPEG-4 AAC explora as redundâncias perceptuais de um sinal de áudio alcançando baixas taxas de bits (BAUER, 2005). Desta forma, com uma taxa de 64kbps o codificador AAC consegue obter um qualidade perceptual igual a de um CD (WANG, ZHANG e YAO, 2008).

Para que um modelo possa ser elaborado é necessário conhecer muito bem a dinâmica efetiva do processo que se deseja modelar. Desta forma, a Figura 29 apresenta a sequência de acontecimentos internos ao codificador MPEG-4 AAC que é abordado e modelado neste trabalho:

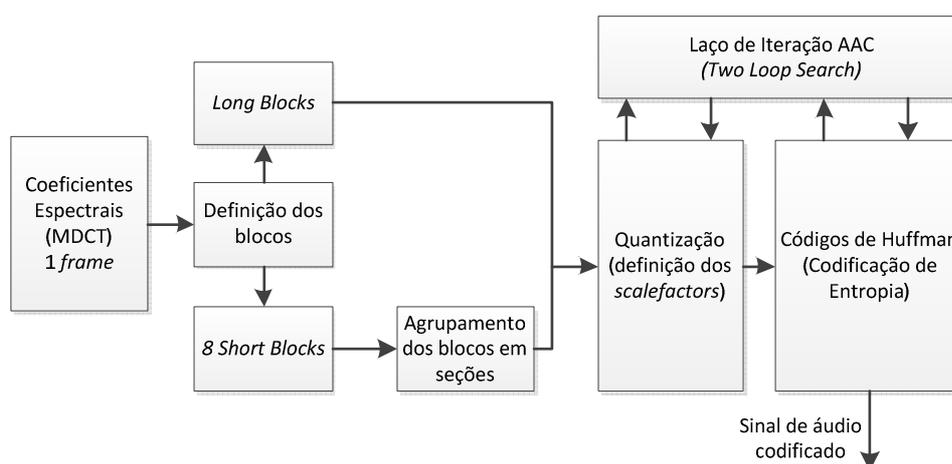


Figura 29 – Dinâmica do processo de codificação de áudio MPEG-4-AAC

A dinâmica do codificador MPEG-4 AAC inicia com a transformação dos dados do sinal de áudio no domínio do tempo, para o domínio da frequência utilizando o método *MDCT*. Após, os coeficientes de um *frame* do sinal são divididos em blocos, podendo ser um *long block* ou 8 *short blocks*. Caso os coeficientes do *frame* sejam divididos em 8 *short blocks*, eles ainda podem ser combinados em seções para diminuir a taxa total de bits. Dentro de um *long block* ou das seções definidas, os coeficientes espectrais são divididos novamente, porém no domínio da frequência, em *scalefactor bands (SFBs)*. Para cada *SFB* o codificador define uma quantidade de bits para representar os coeficientes de frequência. Esta decisão

ocorre dinamicamente, em tempo real, através da iteração do laço de controle de taxa de bits e do laço de controle de distorção (*TLS*). Estes laços definem qual é o intervalo de quantização (*scalefactors*) e qual o código de Huffman que melhor configuram o codificador para uma *scalefactor band* de um *frame*. Além disso, é necessário que o codificador transmita em seu fluxo de dados o valor dos *scalefactors* e dos Códigos de Huffman associados à cada *scalefactor band*, como informações auxiliares. O codificador utiliza também alguns bits, para informar ao decodificador qual o agrupamento utilizado nas seções dos blocos. Desta forma, a taxa total de bits necessária para codificar um determinado sinal de áudio, é a soma dos bits necessários para a codificação das seções e dos blocos, com a soma dos bits necessários para codificar os dados auxiliares. Assim definem-se os quatro parâmetros de codificação que o codificador deve gerenciar (BAUER, 2005):

- 1) Estrutura do bloco (*short block, long block*);
- 2) Agrupamento dos blocos (apenas para *short blocks*);
- 3) Intervalo de quantização (*scalefactors*);
- 4) Codificação de Entropia (Códigos de Huffman - *HCB*).

Para garantir uma baixa taxa de bits, o codificador deve escolher uma boa combinação dos quatro parâmetros, de maneira que a taxa pré-definida não seja ultrapassada, e que ainda garanta uma qualidade perceptual do sinal de áudio. Para medir a qualidade perceptual do sinal codificado, utiliza-se a medida *ANMR* (*Average Noise-to-Mask Ratio*).

Nos trabalhos (BAUER, 2005) (BAUER e VINTON, 2006) (BAUER, FELLERS e DAVIDSON, 2006), os autores definem o processo de otimização de *ANMR* como o Problema de Codificação AAC. Este problema se assemelha com um problema clássico de otimização combinatória, o problema de Seleção de Partes. O problema de Seleção de Partes pertence à classe de problemas de complexidade NP-Completo (JEON e LEEP, 2005). Consiste no agrupamento das partes, baseado em suas similaridades, como por exemplo, a forma geométrica, o processo associado, material de fabricação, entre outros (KUSIAK, DORF e RICHARD, 1994) (SRIVASTAVA e CHEN, 1993). De forma análoga, o Problema de Codificação AAC também possui a característica de agrupamento e seleção de partes, pois deve escolher os *scalefactors*, os *HCBs*, e agrupá-los da melhor maneira dentro das

scalefactors bands para minimizar a relação ruído-máscara *ANMR*. Assim, a complexidade do Problema de Codificação AAC pode ser definida como NP-Completo.

Encontrar a solução ótima do Problema de Codificação AAC é um problema de otimização complexo, devido às interdependências das quatro variáveis de decisão listadas (BAUER, 2005). As abordagens encontradas na literatura sobre a resolução do Problema de Codificação AAC, não permitem implementações em tempo real. Desta forma, o problema acaba sendo simplificado, e ao contrário de realizar uma otimização utilizando os quatro parâmetros de codificação descritos, o Problema de Codificação AAC é simplificado de maneira significativa, negligenciando a relação entre alguns parâmetros, como por exemplo, a interdependência dos *scalefactors* adjacentes abordada na seção 3.2.5.

O método sugerido pela norma (ISO/IEC, 2004) para resolver o Problema de Codificação AAC é a utilização do laço iterativo *TLS* (*Two Loop Search*) descrito na seção 3.2.6. No entanto, algumas desvantagens importantes do procedimento de *TLS* são que ele não necessariamente converge, ignora as correlações inter-banda dos *scalefactors* e dos Códigos de Huffman, e não visa minimizar a distorção total em função de considerar sua análise separadamente por bandas (ALI e SALEEM, 2008) (WANG, ZHANG e YAO, 2008).

O modelo matemático utilizado neste trabalho, para resolver o Problema de Codificação AAC utilizando as metaheurísticas Algoritmo Genético e Busca Tabu em uma estrutura híbrida, baseia-se na abordagem de Claus Bauer (2005). Porém, as metaheurísticas atuaram sobre três variáveis de decisão do problema: os intervalos de quantização (*SF*) e os códigos de Huffman (*HCB*) para cada *scalefactor band* (*SFB*), e o agrupamento dos blocos para cada *frame*, caso este seja configurado como *short block*. O processo de escolha do tipo de bloco, entre *short block* e *long block*, permaneceu sob responsabilidade do codificador MPEG-4 AAC, devido este utilizar a métrica Entropia Perceptual (JOHNSTON, 1988), que identifica a mínima quantidade de bits necessária para a representação do *frame*. Assim, o objetivo das metaheurísticas foi encontrar a melhor combinação entre os três parâmetros listados, para uma determinada taxa de bits pré-definida, de maneira que a qualidade perceptual do áudio pudesse ser mantida.

6.1 Arquitetura do Modelo

Para que a estrutura de simulação desenvolvida para a otimização da qualidade perceptual de áudio pudesse atuar sobre os parâmetros internos do codificador MPEG-4 AAC, a sua arquitetura principal teve que ser alterada. Desta forma, o bloco de Controle Iterativo da Taxa de bits e Distorção (*TLS*) do codificador de referência, foi substituído pela estrutura de otimização implementada utilizando as metaheurísticas. Na Figura 30 pode-se verificar de que maneira esta estrutura foi integrada ao codificador MPEG-4 AAC.

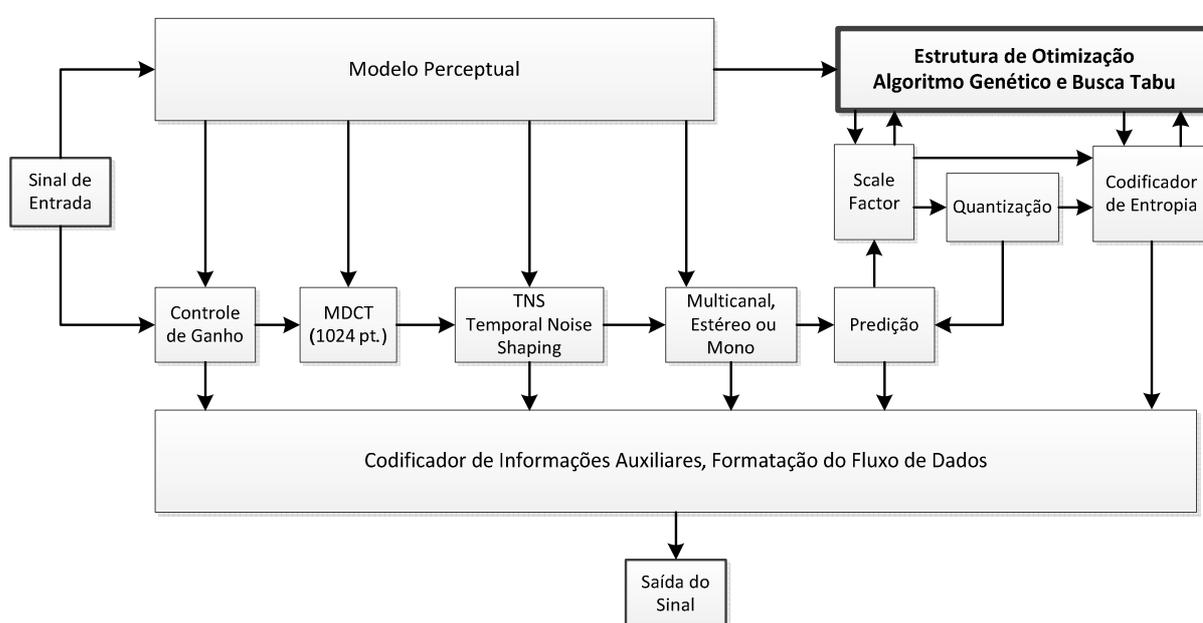


Figura 30 – Simulação do codificador AAC com estrutura de otimização proposta

Desta maneira, o módulo de otimização desenvolvido baseado nas metaheurísticas foi responsável pela definição dos três parâmetros internos do codificador MPEG-4 AAC: os valores dos *scalefactors*, os valores dos Códigos de Huffman e o agrupamento dos blocos (para um *frame* configurado como *short block*). Esta dinâmica de iterações ocorreu em tempo real, de maneira que a cada *frame*, a estrutura das metaheurísticas foi solicitada pelo codificador de áudio, para definição de seus parâmetros.

Na Figura 31 é apresentada a arquitetura interna da estrutura de otimização. Inicialmente, a partir das componentes de frequência e do modelo perceptual, o

codificador realiza o cálculo da Entropia Perceptual, para definir que tipo de configuração de bloco representa melhor o *frame* em análise, oito *short blocks* ou um *long block*. Estabelecida a configuração dos blocos, as metaheurísticas definem através de seus processos de busca, os três parâmetros necessários para configurar o codificador. Caso o *frame* seja configurado com oito *short blocks*, estes ainda podem ser agrupados em seções, desta forma, uma escolha de agrupamento deve ser realizada. Depois, na etapa de quantização devem ser escolhidos os *scalefactors* e os códigos de Huffman para cada *scalefactor band* do *frame*. Ao final deste processo, a medida de qualidade perceptual é calculada, e desta maneira, o resultado da configuração dos parâmetros obtidos através da estrutura de otimização das metaheurísticas, pode ser avaliado.

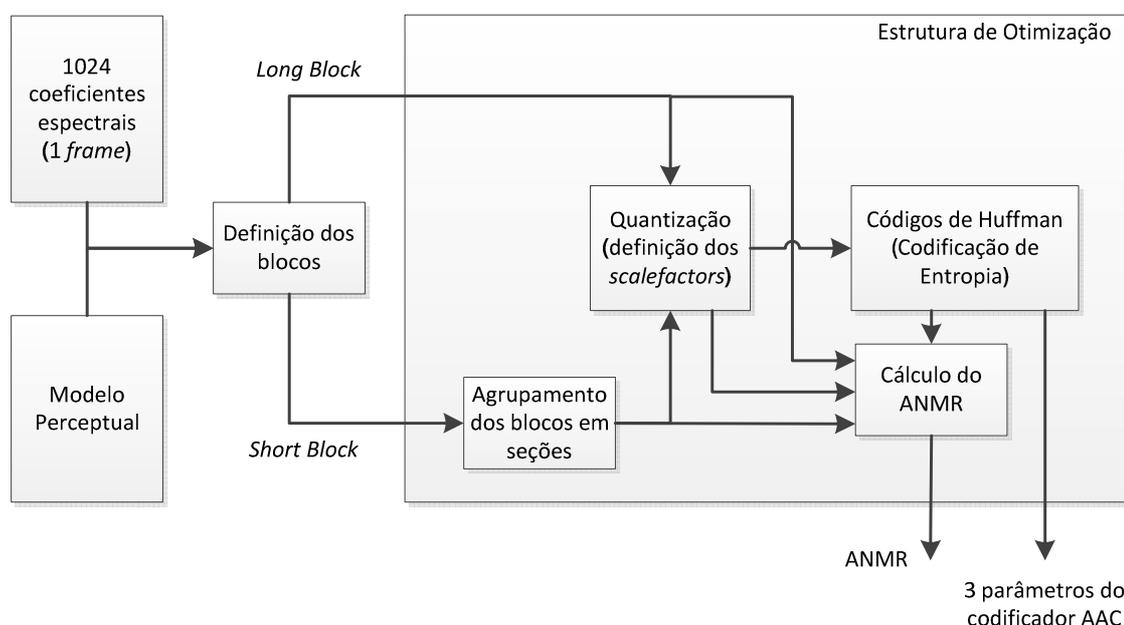


Figura 31 – Arquitetura do modelo proposto

Para a simulação do modelo desenvolvido, foi necessária a utilização do código em C do codificador MPEG-4 AAC desenvolvido pelo projeto de software livre FFmpeg (BELLARD), de maneira que este foi integrado à estrutura de otimização das metaheurísticas.

6.2 Formulação do Problema de Codificação AAC

No codificador MPEG-4 AAC são utilizados 1024 coeficientes espectrais, porém nem todos são relevantes para a percepção sonora. Assim, não há a necessidade de codificá-los individualmente, para isso eles são agrupados em blocos, seções e *scalefactor bands* (*SFBs*) (BAUER e VINTON, 2006). Inicialmente os *frames* com os 1024 coeficientes espectrais são divididos em *long* e *short blocks*. Em um *long block* os coeficientes são divididos tipicamente em $T = 49$ *SFBs* (BAUER, 2005). Este valor pode variar de acordo com a frequência de amostragem do sinal de áudio (ISO/IEC, 2005). Em uma estrutura de *short blocks*, os coeficientes são divididos em oito blocos com 128 coeficientes. Blocos adjacentes podem ser agrupados em seções, de acordo com sua semelhança perceptual. Assim, podem existir de uma até oito seções, que produzem $2^7 = 128$ possíveis escolhas de seções. Cada seção contém $R = 15$ *SFBs*, sendo que o total de *SFBs* dentro de um *frame* é definido por Rq , $1 \leq q \leq 8$, onde q é o número de seções (BAUER, 2005). Desta forma, o número de *SFBs* para *short blocks* e *long blocks*, dentro de um *frame* pode variar conforme a expressão (23).

$$R < T < 8R \quad (23)$$

Uma estrutura de agrupamento dos *short blocks* em seções será denominada de *configuração*. Assim, o número g de configurações pode variar de $g = 0$ até $g = G + 1$, onde as configurações $g = 0, \dots, G = 127$, correspondem às configurações das seções dos *short blocks*, e $g = G + 1 = 128$ corresponde à configuração de um *long block* (BAUER, FELLERS e DAVIDSON, 2006).

Os coeficientes espectrais que pertencem a uma mesma *SFB* são codificados utilizando o mesmo intervalo de quantização, definidos pelos *scalefactors* (*SF*), e o mesmo código de Huffman (*HCB*) que configura o codificador de entropia (BAUER, 2005). Os *SFs* podem assumir no máximo o valor $M_1 = 120$, enquanto que os *HCB* podem variar até $M_2 = 11$ (AGGARWAL, REGUNATHAN e ROSE, 2000).

São atribuídos s_i e h_i para a i -ésima *SFB* do *frame*, os valores de *SF* e *HCB* respectivamente. Os valores s_i e h_i devem ser inteiros e pertencerem à faixa $60 \leq s_i \leq M_1$, $0 \leq h_i \leq M_2$ respectivamente, sendo que i deve pertencer ao intervalo

$1 \leq i \leq N$, onde N é o número de *SFBs*. Assim, os valores dos *SFs* e *HCBs* podem ser expressos em vetores $S = \{s_1, \dots, s_N\}$ e $H = \{h_1, \dots, h_N\}$ (BAUER, 2005).

Matematicamente a medida *ANMR* é modelada conforme a equação (24) (BAUER, 2005):

$$ANMR(S, g) = \frac{1}{N} \sum_{i=1}^N w_{g,i} \cdot d(s_i) \quad (24)$$

Na expressão (24), o erro de quantização é definido por $d(s_i)$ para a i -ésima *SFB*, se o i -ésimo *SF* é escolhido igual a s_i . O inverso do limiar de mascaramento que define o comportamento do ouvido humano (Figura 8), é definido pelo peso $w_{g,i}$, da i -ésima *SFB* na g -ésima configuração. Na Figura 32 é apresentado um diagrama em blocos da modelagem da medida *ANMR* expressa na equação (24).

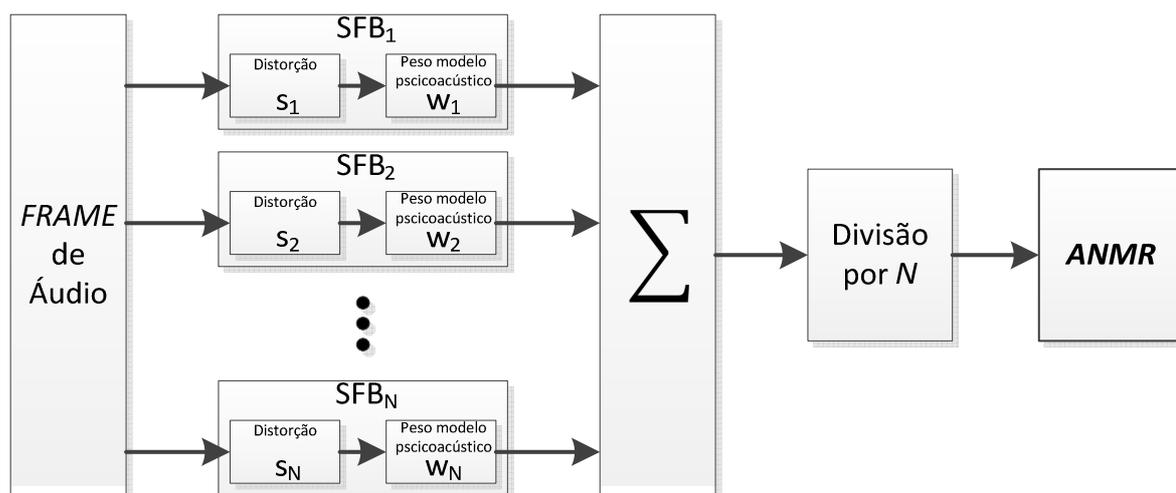


Figura 32 – Representação da modelagem da métrica *ANMR*

A taxa de transmissão é a principal restrição do modelo do codificador AAC, pois uma vez determinada pelo usuário, esta taxa de bits não deve ser ultrapassada. A fim de identificar as primeiras *SFBs* de cada *long block*, e de cada seção, foi criada a notação 1_g . Assim é dito que $i \in 1_g$ se:

$$(i = 1 \vee g = G + 1) \wedge (SFB_i \text{ é a primeira na seção de configuração } g \vee g \leq G).$$

A modelagem da taxa de transmissão consiste em cinco partes (BAUER, 2005):

- Se o *frame* for configurado como *short block*, o número de bits B necessário para informar ao decodificador sobre a decisão das seções é proporcional ao número de seções q , conforme a equação (25):

$$B = c(q - 1), \quad (25)$$

onde c é uma constante.

Caso o *frame* seja configurado como um *long block*, então $B = 0$.

- $Q_i(s_i, h_i, g)$ é a quantidade de bits necessários para quantizar os coeficientes espectrais, da i -ésima *SFB* com os valores de *SF* e *HCB* escolhidos como s_i e h_i respectivamente, para uma configuração g de blocos e seções.
- A função $F(s_{i-1}, s_i)$ representa o número de bits necessários para codificar o *SF* da i -ésima *SFB*, como informação auxiliar dentro do fluxo de dados do codificador. Caso $i \notin 1_g$, ou seja, não seja o primeiro elemento de uma *SFB*, então os *SF* são codificados diferencialmente, de maneira que $F(s_{i-1}, s_i) = F(s_i - s_{i-1})$. Porém se $i \in 1_g$, ou seja, se trata do primeiro *SFB*, então não há como codificar *SF* diferencialmente, e assim a quantidade de bits necessários para codificar s_i é expressa por $F(s_i)$.
- A expressão $G_i(H, g)$ representa o número de bits necessários para codificar o valor de *HCB* da i -ésima *SFB* na g -ésima configuração. Se $i \notin 1_g$ e $h_i = h_{i-1}$, então $G_i(H, g) = 0$, ou seja, se o valor de *HCB* atual for igual ao seu adjacente anterior, então não há a necessidade de codificar este mesmo valor. Porém se $h_i \neq h_{i-1}$, então o valor de $G_i(H, g)$ depende do comprimento da seção *HCB* partindo de *SFB* _{i} . Uma seção *HCB* é definida como uma sequência de *SFB* _{i} que são codificadas com um mesmo *HCB*. Assim, dado o intervalo $a \leq i \leq b$ de uma seção *HCB*, o seu respectivo tamanho é definido pela diferença entre o término da seção *SFB* _{b} , e seu início *SFB* _{a} , de maneira que o comprimento $l(a) = b - a + 1$.

Para uma configuração *long block* $g = G + 1$, e $1 \leq i \leq T$,

$$G_i(H, G + 1) = \begin{cases} 0 \text{ bit,} & \text{se } i \notin 1_g, h_i = h_{i-1}; \\ 9 \text{ bits,} & \text{se } i \in 1_g, l(h_i) \leq 30, \text{ ou } i \notin 1_g, h_i \neq h_{i-1}, l(h_i) \leq 30; \\ 13 \text{ bits,} & \text{se } i \in 1_g, l(h_i) \geq 31, \text{ ou } i \notin 1_g, h_i \neq h_{i-1}, l(h_i) \geq 31. \end{cases}$$

Para *short blocks* com configurações $g \leq G$, a quantidade de bits em função do tamanho de uma seção *HCB* é descrita por:

$$G_i(H, g) = \begin{cases} 0 \text{ bits,} & \text{se } i \notin 1_g, h_i = h_{i-1}; \\ 7 \text{ bits,} & i \in 1_g, l(h_i) \leq 6, \text{ ou } i \notin 1_g, h_i \neq h_{i-1}, l(h_i) \leq 6; \\ 10 \text{ bits,} & \text{se } i \in 1_g \text{ ou } i \notin 1_g, 7 \leq l(h_i) \leq 13, h_i \neq h_{i-1}, 7 \leq l(h_i) \leq 13; \\ 13 \text{ bits,} & \text{se } i \in 1_g \text{ ou } i \notin 1_g, l(h_i) \geq 14, h_i \neq h_{i-1}, l(h_i) \geq 14. \end{cases}$$

- Caso $i \in 1_g$, ou seja, se trata da primeira *SFB* de um *long block* ou de uma seção, o cálculo da contribuição das *SFBs* para a taxa total de bits é realizada de maneira diferente, quando comparada às demais *SFBs*, caso $i \notin 1_g$. Para $i \notin 1_g$ a quantidade de bits depende da diferença entre os valores de s_i e h_i , o que não ocorre para $i \in 1_g$. Desta forma, para o caso de $i \in 1_g$, as contribuições de bits serão denotadas apenas por $F(\cdot)$, $G(\cdot)$, conforme a expressão (26):

$$Q_i(s_i, h_i, g) \rightarrow Q_i(s_i, h_i, g) + F(s_i) + G_i(H, g) \quad (26)$$

Desta forma, é possível definir a taxa de bits gerada no processo de codificação $R(S, H, g, B)$ através da equação (27) (BAUER, 2005):

$$R(S, H, g, B) = \sum_{i=1}^N Q_i(s_i, h_i, g) + \sum_{\substack{i=1 \\ i \notin 1_g}}^N [F(s_i - s_{i-1}) + G_i(H, g)] + B, \quad (27)$$

onde $N \in \{T, Rq\}$, $1 \leq q \leq 8$ é o número de bandas (*SFBs*) de uma configuração g . Assim, para um determinado limite de taxa de bits R_t , e a medida de qualidade perceptual $ANMR(S, g)$ descrita pela expressão (24), o Problema de Codificação AAC pode ser definido através das expressões (28) e (29) (BAUER, 2005):

$$\text{Minimize} \quad ANMR(S, g) \quad (28)$$

$$\text{Sujeito a} \quad R(S, H, g, B) \leq R_t \quad (29)$$

Desta forma, o modelo é estruturado de maneira a minimizar a relação $ANMR$, evitando que a taxa de bits pré-definida R_t , seja ultrapassada. Para uma

representação deste modelo em uma arquitetura computacional, a formulação matemática deve ser aprofundada. Assim, o modelo matemático será transformado em um modelo computacional equivalente.

6.2.1 Formulação Matemática

Para melhor definir e facilitar a construção do modelo computacional do Problema de Codificação AAC, será considerada a existência de $8R$ *SFBs* dentro de um *frame* para uma configuração g . Porém, para que o número de *SFBs* possa variar dentro de um *frame*, será introduzido o conceito de *SFB* válida caso esta exista dentro da configuração g , e *SFB* inválida caso não exista para a configuração g (BAUER, FELLERS e DAVIDSON, 2006).

Caso uma *SFB* seja definida como inválida, então ela não afeta as expressões da Função Objetivo e das restrições. Para uma configuração $g = G + 1$, que caracteriza um *long block*, as primeiras T *SFBs* são válidas, enquanto que as demais $8R - T$ bandas são inválidas.

Uma seção que é definida por SB_j , $1 \leq j \leq 8$, é uma combinação entre *short blocks* adjacentes de um *frame*, em função de sua semelhança perceptual. Assim, como um *short block* possui R *SFBs*, cada SB_j também irá possuir R *SFBs*.

A fim de representar de que maneira os *short blocks* são agrupados formando as seções, foi utilizada uma variável binária auxiliar para fazer esta codificação. Assim, a variável u_j , $0 \leq j \leq 7$ indica que os *short blocks* SB_j e SB_{j+1} pertencem a mesma seção se $u_j = 0$. Caso $u_j = 1$, então SB_j e SB_{j+1} não pertencem a mesma seção. Devido um conjunto de 8 *short blocks* ter no mínimo uma seção, u_0 sempre será igual a 1. Utilizando esta codificação de u_j é criada uma relação entre todos os *short blocks* de todas as possíveis configurações g , com todas as escolhas de u_j . Ainda deve-se observar que as bandas (*SFBs*) da seção SB_j são válidas se $u_j = 1$ e inválidas se $u_j = 0$. Assim, para um *short block* com uma configuração de s seções, $1 \leq s \leq 8$, existem s *short blocks* SB_{a_j} , $1 \leq j \leq s$, $1 \leq a_1 < \dots < a_j < \dots < a_s \leq 8$ com bandas válidas. As *SFBs* de uma seção SB_{a_j} correspondem à *SFB* da j -ésima das s seções. As outras *SFBs*, contidas em $8 - s$ *SBs*, não são válidas. Desta forma, para

uma configuração g , o conjunto de bandas válidas é N_g (BAUER, 2005). Assim, sendo a codificação das seções de um *frame* composto de 8 *short blocks* expressa por $u = 10110101$, o agrupamento pode ser visualizado na Figura 33:

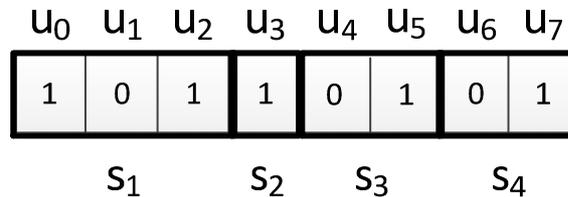


Figura 33 – Codificação do agrupamento de *short blocks*

A partir da codificação da Figura 33 é possível identificar quais *short blocks* e *SFBs* pertencem a cada seção. Esta interpretação é apresentada na Tabela 7.

Seções	Número de <i>shortblocks</i>	<i>SFBs</i> válidas
Seção 1	3	R = 15
Seção 2	1	R = 15
Seção 3	2	R = 15
Seção 4	2	R = 15

Tabela 7 – Interpretação do agrupamento dos *short blocks* em seções

No exemplo da Figura 33 e da Tabela 7 é possível identificar que através da codificação de u existem quatro seções com 15 bandas (*SFBs*) cada uma. Isso significa que este *frame* foi codificado utilizando 60 *SFBs* no total.

Para facilitar a manipulação de $G_i(H, g)$ dentro do modelo computacional, que representa a quantidade de bits em função do tamanho de uma seção *HCB*, esta função foi desmembrada em um conjunto de outras funções. Inicialmente foi considerada a diferença a dos valores de *HCB* de duas *SFBs* adjacentes $a = h_i - h_{i-1}$ (BAUER, FELLERS e DAVIDSON, 2006). Para *frames* configurados como *long blocks* define-se a função $G_L(a)$ e a variável x_1 :

$$G_L(a) = \begin{cases} 0 \text{ bit}, & a = 0, \\ 9 \text{ bits}, & a \neq 0. \end{cases}$$

$$x_1 = \begin{cases} 1, & \text{se } g = G + 1 \text{ e a seção HCB possui tamanho } \geq 31 \text{ e está entre } SFB_1 \text{ e } SFB_T \\ 0, & \text{caso contrário.} \end{cases}$$

Nas sentenças acima verifica-se que se não há uma seção de *HCB* de tamanho ≥ 31 então $G_i(H, 128) = G_L(h_i - h_{i-1})$ para todo $2 \leq i \leq T = 49$. Para a seção de *HCB* com tamanho ≥ 31 , são necessários 13 bits para sua representação e assim a quantidade de bits pode ser expressa por $G_i(H, 128) = G_L(h_i - h_{i-1}) + 4x_1$.

Para configurações de *frames* $g \leq G$ que utilizam *short blocks*, as definições da quantidade de bits das seções *HCB* são diferentes das que utilizam *long blocks* (BAUER, 2005):

$$G_S(a) = \begin{cases} 0 \text{ bit}, & a = 0, \\ 7 \text{ bits}, & a \neq 0. \end{cases}$$

$$x_{2,j} = \begin{cases} 3, & \text{se } u_j = 1, g \leq G \text{ e uma seção HCB possui tamanho } \geq 7 \\ & \text{e está entre } SFB_{R_j+1} \text{ e } SFB_{R(j+1)}; \\ 0, & \text{caso contrário.} \end{cases}$$

$$x_{3,j} = \begin{cases} 3, & \text{se } u_j = 1, g \leq G \text{ e duas seções HCB possuem tamanho } \geq 7 \\ & \text{e está entre } SFB_{R_j+1} \text{ e } SFB_{R(j+1)}; \\ 0, & \text{caso contrário.} \end{cases}$$

$$x_{4,j} = \begin{cases} 3, & \text{se } u_j = 1, g \leq G \text{ e uma seção HCB possui tamanho } \geq 14 \\ & \text{e está entre } SFB_{R_j+1} \text{ e } SFB_{R(j+1)}; \\ 0, & \text{caso contrário.} \end{cases}$$

Através das notações apresentadas, verifica-se que se $u_j = 0$, então as variáveis $x_{i,j}$, para $2 \leq i \leq 4$ e $0 \leq j \leq 7$ são iguais a zero, o que determinam que as seções SB_j dos *short blocks* contém apenas SFB_i inválidas. Não havendo uma seção de *HCB* com tamanho ≥ 7 , e $u_j = 1$ então $G_i(H, g) = G_S(h_i - h_{i-1})$ para todo $R_j + 2 \leq i \leq R(j + 1)$. Existem ainda outras três situações em que uma ou duas seções *HCB* possuem tamanho entre 7 e 13 ou uma seção com tamanho ≥ 14 . Uma análise separadamente de cada uma destas situações mostra que para a SFB_i onde a seção *HCB* inicia, a quantidade de bits para codificar uma seção *HCB* é $G_i(H, g) = G_S(h_i - h_{i-1}) + r$, onde (BAUER, FELLERS e DAVIDSON, 2006):

- $r = 3 = 3x_{2,j}$, se a primeira seção *HCB* de tamanho entre 7 e 13 inicia na SFB_i ;
- $r = 6 = 3x_{3,j}$, se a segunda seção *HCB* de tamanho entre 7 e 13 inicia na SFB_i ;
- $r = 6 = 3(x_{2,j} + x_{4,j})$, se a seção *HCB* de tamanho ≥ 14 inicia na SFB_i ;

Através das definições realizadas, é possível obter a equação (30) que define a quantidade de bits necessários para representar as seções dos *HCBs* em função do seu tamanho.

$$\sum_{\substack{i=1 \\ i \in N_g}}^N G_i(H, g) = \sum_{i=2}^T G_L(h_i - h_{i-1}) + 4x_1 + \sum_{\substack{i=1 \\ i \in N_g, i \notin 1_g}}^{8R} G_S(h_i - h_{i-1}) + \sum_{j=1}^8 3(x_{2,j} + x_{3,j} + x_{4,j}) \quad (30)$$

O primeiro somatório do lado direito da igualdade descreve a quantidade de bits das seções *HCBs* caso o *frame* seja codificado como *long block*. Os outros dois somatórios modelam a quantidade de bits, caso o *frame* seja codificado como *short block*.

De maneira análoga ao que foi realizado para os valores de *HCB*, as funções para os valores de *SF* também devem ser adequadas para construção do modelo computacional. Assim, para uma determinada configuração g de um *frame short block*, a variável $A_{S,i,a}$ descreve a diferença entre os valores de *scalefactors* s_i e s_{i-1} . Caso SFB_i seja uma banda válida, a variável $A_{S,i,a}$ será igual a 1 se $s_i - s_{i-1} = a - M_1$. Esta condição faz com que os valores de *scalefactors*, nunca sejam negativos, devido ao deslocamento na expressão em $-M_1$ unidades. Desta forma, os valores de a estão contidos no intervalo de $1 \leq a \leq 2M_1 - 1$. Caso $s_i - s_{i-1} \neq a - M_1$ então será atribuído o valor zero para $A_{S,i,a}$. O valor de $A_{S,i,a}$ é igual a zero para todos os possíveis valores de a caso a SFB_i seja inválida. De maneira análoga, a variável $B_{S,i,b}$ descreve a diferença entre os valores h_i e h_{i-1} do código de Huffman. Assim, para um *frame long block* e um *frame short block* pode-se expressar:

- *Scalefactors* para *long block*:

$$A_{L,i,a} = \begin{cases} 1, & \text{se } g = G + 1, \text{ e } s_i - s_{i-1} = a - M_1, \\ 0, & \text{caso contrário} \end{cases},$$

para $2 \leq i \leq T$ e $1 \leq a \leq 2M_1 - 1$.

- Códigos de Huffman para *long block*:

$$B_{L,i,b} = \begin{cases} 1, & \text{se } g = G + 1, \text{ e } h_i - h_{i-1} = b - M_2, \\ 0, & \text{caso contrário} \end{cases},$$

para $2 \leq i \leq T$ e $1 \leq b \leq 2M_2 - 1$.

- *Scalefactors* para *short block*:

$$A_{S,i,a} = \begin{cases} 1, & \text{se } 0 \leq g \leq G, \text{ SFB}_i \text{ é válida e } s_i - s_{i-1} = a - M_1, \\ 0, & \text{caso contrário} \end{cases},$$

para $2 \leq i \leq 8R$ e $1 \leq a \leq 2M_1 - 1$.

- Códigos de Huffman para *short block*:

$$B_{S,i,b} = \begin{cases} 1, & \text{se } 0 \leq g \leq G, \text{ SFB}_i \text{ é válida e } h_i - h_{i-1} = b - M_2, \\ 0, & \text{caso contrário} \end{cases},$$

para $2 \leq i \leq 8R$ e $1 \leq b \leq 2M_2 - 1$.

Define-se também a notação de $Z_{g,i,a,b}$, $\forall i, 1 \leq i \leq N$, $\forall a, 1 \leq a \leq M_1$, $\forall b, 1 \leq b \leq M_2$, que representa os valores de s_i e h_i para a i -ésima *SFB* da g -ésima configuração (BAUER, 2005):

$$Z_{g,i,a,b} = \begin{cases} 1, & \text{se } s_i = a, h_i = b \text{ e a configuração } g \text{ é escolhida,} \\ 0, & \text{caso contrário} \end{cases}$$

O peso da distorção da i -ésima banda (*SFB*), quando a configuração g é escolhida, pode ser representado por $e_{i,m}$:

$$e_{i,m} = w_{g,i} d(s_i = m), \text{ para } 1 \leq i \leq 8R, 1 \leq m \leq M_1,$$

onde $w_{g,i}$ é definido como o nível de importância para o ouvido humano da i -ésima SFB na g -ésima configuração, e $d(s_i = m)$ é o erro de quantização provocado pelo *scalefactor* ao ser definido como m .

Para descrição do modelo computacional as variáveis foram abreviadas $Z_{g,i,a,b}$ como Z , $A_{L,i,a}$ e $A_{S,i,a}$ como A , $B_{L,i,b}$ e $B_{S,i,b}$ como B , u_j como U e o conjunto de variáveis $x_1, x_{2,j}, x_{3,j}, x_{4,j}$ como X .

Desta forma, a partir da formulação matemática exposta é possível apresentar o modelo matemático que foi utilizado na representação do modelo computacional para o Problema de Codificação AAC abordado neste trabalho. A equação (31), que expressa a Função Objetivo, e as equações de (32) a (63) que representam as restrições, definem a formulação matemática do problema em estudo (BAUER, FELLERS e DAVIDSON, 2006).

Minimize

$$ANMR(Z) = \frac{1}{N} \sum_{g=0}^{G+1} \sum_{i=1}^{8R} \sum_{a=1}^{M_1} \sum_{b=1}^{M_2} e_{g,i,a} Z_{g,i,a,b}; \quad (31)$$

tal que (32) até (63) sejam satisfeitas, onde:

$$R(Z, A, B, g, U, X) \leq R_t \quad (32)$$

$$\begin{aligned} R(Z, A, B, g, U, X) = & \sum_{g=0}^{G+1} \sum_{i=1}^{8R} \sum_{a=1}^{M_1} \sum_{b=1}^{M_2} Z_{g,i,a,b} Q_{g,i}(a, b) \\ & + \sum_{i=2}^T \sum_{a=0}^{2M_1} A_{L,i,a} F^*(a) + \sum_{i=2}^{8R} \sum_{a=0}^{2M_1} A_{S,i,a} F^*(a) + \sum_{i=2}^T \sum_{b=0}^{2M_2} B_{L,i,b} G^*(b) \\ & + \sum_{i=2}^{8R} \sum_{b=0}^{2M_2} B_{S,i,b} G^*(b) + 4x_1 + 3 \sum_{j=0}^7 (x_{2,j} + x_{3,j} + x_{4,j}) + c \sum_{j=1}^7 u_j \end{aligned} \quad (33)$$

$$\sum_{g=0}^{G+1} v_g = 1 \quad (34)$$

$$\sum_{j=1}^7 u_j 2^{j-1} = \sum_{g=0}^G g v_g \quad (35)$$

$$\sum_{a=1}^{M_1} \sum_{b=1}^{M_2} Z_{g,i,a,b} = v_g, \quad 0 \leq g \leq G+1, \quad i \in N_g \quad (36)$$

$$\sum_{a=1}^{M_1} \sum_{b=1}^{M_2} Z_{g,i,a,b} = 0, \quad 0 \leq g \leq G+1, \quad i \notin N_g \quad (37)$$

$$\sum_{a=1}^{2M_1-1} A_{L,i,a} \leq 1, \quad \forall i \quad 2 \leq i \leq T \quad (38)$$

$$\sum_{b=1}^{2M_2-1} B_{L,i,b} \leq 1, \quad \forall i \quad 2 \leq i \leq T \quad (39)$$

$$\sum_{a=1}^{2M_1-1} A_{S,i,a} \leq 1, \quad \forall i \quad 2 \leq i \leq 8R, \quad i \neq 1_g \quad (40)$$

$$\sum_{b=1}^{2M_2-1} B_{S,i,b} \leq 1, \quad \forall i \quad 2 \leq i \leq 8R, \quad i \neq 1_g \quad (41)$$

$$\sum_{i=2}^T \sum_{a=1}^{2M_1-1} A_{L,i,a} = (T-1)v_{G+1}, \quad 0 \leq j \leq 7 \quad (42)$$

$$\sum_{i=2}^T \sum_{b=1}^{2M_2-1} B_{L,i,b} = (T-1)v_{G+1}, \quad 0 \leq j \leq 7 \quad (43)$$

$$\sum_{i=Rj+2}^{R(j+1)} \sum_{a=1}^{2M_1-1} A_{S,i,a} = (R-1)u_j, \quad 0 \leq j \leq 7 \quad (44)$$

$$\sum_{i=Rj+2}^{R(j+1)} \sum_{b=1}^{2M_2-1} B_{S,i,b} = (R-1)u_j, \quad 0 \leq j \leq 7 \quad (45)$$

$$A_{L,i,a} + 2 - v_{G+1} \geq \sum_{d_3=1}^{M_2} Z_{G+1,i-1,c,d_3} + \sum_{d_4=1}^{M_2} Z_{G+1,i,c+a-M_1,d_4}, \quad 2 \leq i \leq T \quad (46)$$

$$B_{L,i,b} + 2 - v_{G+1} \geq \sum_{d_5=1}^{M_1} Z_{G+1,i-1,d_5,d} + \sum_{d_6=1}^{M_1} Z_{G+1,i,d_6,d+b-M_2}, \quad 2 \leq i \leq T \quad (47)$$

$$A_{S,i,a} + 2 - u_j \geq \sum_{g=1}^G \sum_{d_3=1}^{M_2} Z_{g,i-1,c,d_3} + \sum_{g=1}^G \sum_{d_4=1}^{M_2} Z_{g,i,c+a-M_1,d_4} \quad (48)$$

$$B_{S,i,b} + 2 - u_j \geq \sum_{g=1}^G \sum_{d_5=1}^{M_1} Z_{g,i-1,d_5,d} + \sum_{g=1}^G \sum_{d_6=1}^{M_1} Z_{g,i,d_6,d+b-M_2} \quad (49)$$

Para:

$$0 \leq j \leq 7;$$

$$Rj + 2 \leq i \leq R(j + 1);$$

$$1 \leq a \leq 2M_1 - 1;$$

$$1 \leq b \leq 2M_2 - 1;$$

$$\max(1, 1 + M_1 - a) \leq c \leq \min(M_1, 2M_1 - a);$$

$$\max(1, 1 + M_2 - b) \leq d \leq \min(M_2, 2M_2 - a).$$

$$t_k \geq \frac{1}{30} \sum_{i=k}^{k+29} B_{L,i,M_2} - 1 \quad (50)$$

$$Tx_1 \geq \sum_{k=2}^{T-29} t_k \quad (51)$$

$$g_k \geq \frac{1}{5} \sum_{i=k}^{k+5} B_{S,i,M_2} - 1 \quad (52)$$

$$Rx_{2,j} \geq \sum_{k=Rj+2}^{R(j+1)-6} g_k \quad (53)$$

$$d_{1,j} \geq \frac{1}{11} \sum_{i=Rj+2}^{R(j+1)} B_{S,i,M_2} - 1 \quad (54)$$

$$e_{1,j} \geq 1 - B_{S,Rj+8,M_2} - B_{S,Rj+9,M_2} \quad (55)$$

$$e_{2,j} \geq 1 - B_{S,Rj+8,M_2} - B_{S,Rj+15,M_2} \quad (56)$$

$$e_{3,j} \geq 1 - B_{S,Rj+2,M_2} - B_{S,Rj+9,M_2} \quad (57)$$

$$d_{2,j} \geq \frac{1}{13} \sum_{i=Rj+2}^{R(j+1)} B_{S,i,M_2} - 1 \quad (58)$$

$$e_{4,j} \geq 2 - B_{S,Rj+8,M_2} - B_{S,Rj+9,M_2} \quad (59)$$

$$x_{3,j} \geq \frac{1}{3} (d_{1,j} + e_{1,j} + e_{2,j} + e_{3,j}) - 1 \quad (60)$$

$$x_{3,j} \geq d_{2,j} + e_{4,j} - 1 \quad (61)$$

$$w_k \geq \frac{1}{12} \sum_{i=k}^{i+12} B_{S,i,M_2} - 1 \quad (62)$$

$$Rx_{4,j} \geq \sum_{k=Rj+2}^{R(j+1)-12} w_k \quad (63)$$

para $0 \leq j \leq 7$.

As variáveis Z , X , v_g , u_j , $e_{i,j}$, $d_{i,j}$, g_k , t_k , w_k devem ser inteiras $\in \{0,1\}$, e as variáveis A e B podem assumir apenas valores inteiros positivos.

A partir das restrições listadas acima, deve-se fazer algumas observações. A equação (34) demonstra que somente uma configuração g é escolhida por vez pelo codificador. Para $g \leq G$, a equação (35) define de que forma ou não, os *short blocks* irão ser agrupados em seções. Como o conceito de seção não se aplica a uma

estrutura de *frame* configurado como *long block*, neste caso todos u_j devem ser nulos.

As escolhas dos *scalefactors* (*SFs*) e dos *HCBs* para cada *SFB* são definidos pelas variáveis $Z_{g,i,a,b}$, $A_{i,a}$, $B_{L,i,b}$ e $B_{S,i,b}$. As equações (36) e (37) garantem que para uma escolha de uma configuração g , cada banda válida SFB_i terá apenas uma variável $Z_{g,i,a,b}$ igual a 1. As demais variáveis serão iguais a zero. Para *frames long blocks* as variáveis $Z_{G+1,i,a,b}$ definem os parâmetros. Assim, as equações (38), (42) e (46) garantem que se $i \notin N_g$ e $s_i - s_{i-1} = a - M_1$ então $A_{L,i,a} = 1$ e todas as outras $A_{L,i,a} = 0$. De maneira análoga as equações (39), (43) e (47) possuem a mesma relação com as variáveis $B_{L,i,b}$, que representam a escolha dos códigos de Huffman.

Para $g \leq G$ são atribuídos valores para as variáveis $Z_{g,i,a,b}$, as equações (40), (44) e (48) garantem que para qualquer $u_j = 1$ e $Rj + 2 \leq i \leq R(j + 1)$, se $s_i - s_{i-1} = a - M_1$, então $A_{S,i,a} = 1$ e os demais são $A_{S,i,a} = 0$. As equações (41), (45) e (49) estabelecem as mesmas relações para as variáveis $B_{S,i,b}$. Para todas as bandas inválidas, as variáveis A e B são iguais a zero.

Se dentro de um *long block* existe pelo menos uma seção *HCB* com tamanho ≥ 31 , então as equações (50) e (51) mostram que $x_1 = 1$. Caso não haja pelo menos um *HCB* com tamanho ≥ 31 , então $x_1 = 0$. Deve-se observar que a variável x_1 não representa a escolha do parâmetro *HCB*, mas expressa o impacto sobre a taxa de bits $R(Z, A, B, X)$, de uma determinada escolha.

Para os *frames* configurados como *short blocks*, as equações (52) e (53) forçam as variáveis $x_{2,j} = 1$, se no mínimo existe uma seção *HCB* de tamanho ≥ 7 entre as bandas SFB_{Rj+1} e $SFB_{R(j+1)}$. As equações (55) a (61), fazem com que $x_{3,j} = 1$, se existem duas seções *HCBs* de tamanho ≥ 7 entre as bandas SFB_{Rj+1} e $SFB_{R(j+1)}$. Em uma análise de todas as possíveis escolhas de *HCBs* que contém duas seções *HCBs* de tamanho ≥ 7 , as equações (54) a (57) e (60) modelam o caso de haver duas seções *HCBs* de tamanho 7, e as equações (58) a (59) e (61) modelam o caso quando existe uma seção *HCB* de tamanho 7 e uma seção *HCB* de tamanho 8. Por fim, as equações (62) e (63) fazem $x_{4,j} = 1$, se uma seção *HCB* possui tamanho ≥ 14 .

6.3 Estrutura de Simulação do Codificador MPEG-4 AAC

Nesta seção é abordada a estrutura de simulação desenvolvida para a otimização do codificador MPEG-4 AAC. Desta forma, inicialmente são descritos os métodos e técnicas utilizados na implementação do Algoritmo Genético, e na sequência é detalhado o algoritmo desenvolvido para a Busca Tabu, ambos focados no Problema de Codificação AAC. Ao final da seção é apresentada a estrutura híbrida das metaheurísticas, contemplando a união do Algoritmo Genético e Busca Tabu, em um único Algoritmo Híbrido para a otimização do codificador MPEG-4 AAC.

6.3.1 Algoritmo Genético para o Problema de Codificação AAC

Nesta seção é apresentado o desenvolvimento e implementação da metaheurística Algoritmo Genético aplicado ao Problema de Codificação AAC. São detalhadas a formatação do cromossomo, o processo para a obtenção da população inicial, o processo de cruzamento e mutação dos indivíduos e o critério de parada.

6.3.1.1 Codificação do Cromossomo

As três variáveis de decisão abordadas neste trabalho foram codificadas em uma estrutura de solução de maneira que foram utilizados três vetores: um para representar os *scalefactors*, um para os códigos de Huffman e outro para representar o agrupamento dos blocos em caso de *short blocks*. Informações relevantes para o *frame* como a quantidade de bits utilizada, quantidade de *SFBs* no *frame*, o valor de *ANMR* de cada banda de áudio e total do *frame*, também foram associadas à mesma estrutura da solução. Desta forma, em um único objeto (estrutura) ficaram contidas todas as informações necessárias para a definição dos parâmetros internos do codificador (Figura 34).



Figura 34 – Codificação da solução

Os *SFs* foram representados por um vetor de 120 posições, sendo cada posição uma *scalefactor band* do *frame*. Como a quantidade de *SFBs* pode variar de acordo com a definição dos blocos (*long* ou *short*), o tamanho do vetor foi definido pela maior configuração. De maneira análoga, os códigos de Huffman (*HCBs*) também foram representados por um vetor de 120 posições para descrever as possíveis 120 *SFBs* de um *frame*. O agrupamento das seções dos blocos foi representado por um vetor de oito posições, sendo que cada uma das posições identifica uma das oito seções do *frame*. O número de cada posição do vetor indica de que maneira as seções estão agrupadas. Ao todo são possíveis 127 combinações de agrupamentos das seções dentro de um *frame*. Na Figura 35 é apresentado um exemplo do agrupamento das seções em três grupos:



Figura 35 – Codificação do agrupamento das seções

Um número diferente de zero dentro do vetor de agrupamento indica o início de uma nova seção e também o seu tamanho. Desta forma, no exemplo da Figura 35, o primeiro grupo é formado por três seções, o segundo por duas seções e o terceiro e último grupo por três seções. Deve-se observar que o agrupamento de seções dentro de um *frame* apenas é realizado, caso este seja configurado como *short block*.

6.3.1.2 População Inicial

A população inicial utilizada pelo Algoritmo Genético foi gerada pelo método existente dentro do codificador MPEG-4 AAC, o *Two Loop Search (TLS)*. Porém, este foi alterado para executar apenas uma vez o laço externo, e uma vez o laço interno. Para que um conjunto de distintas soluções pudesse ser gerado, formando a população inicial, a solução gerada pela *TLS* foi suavemente alterada. A dinâmica de geração da população inicial é apresentada na Figura 36.

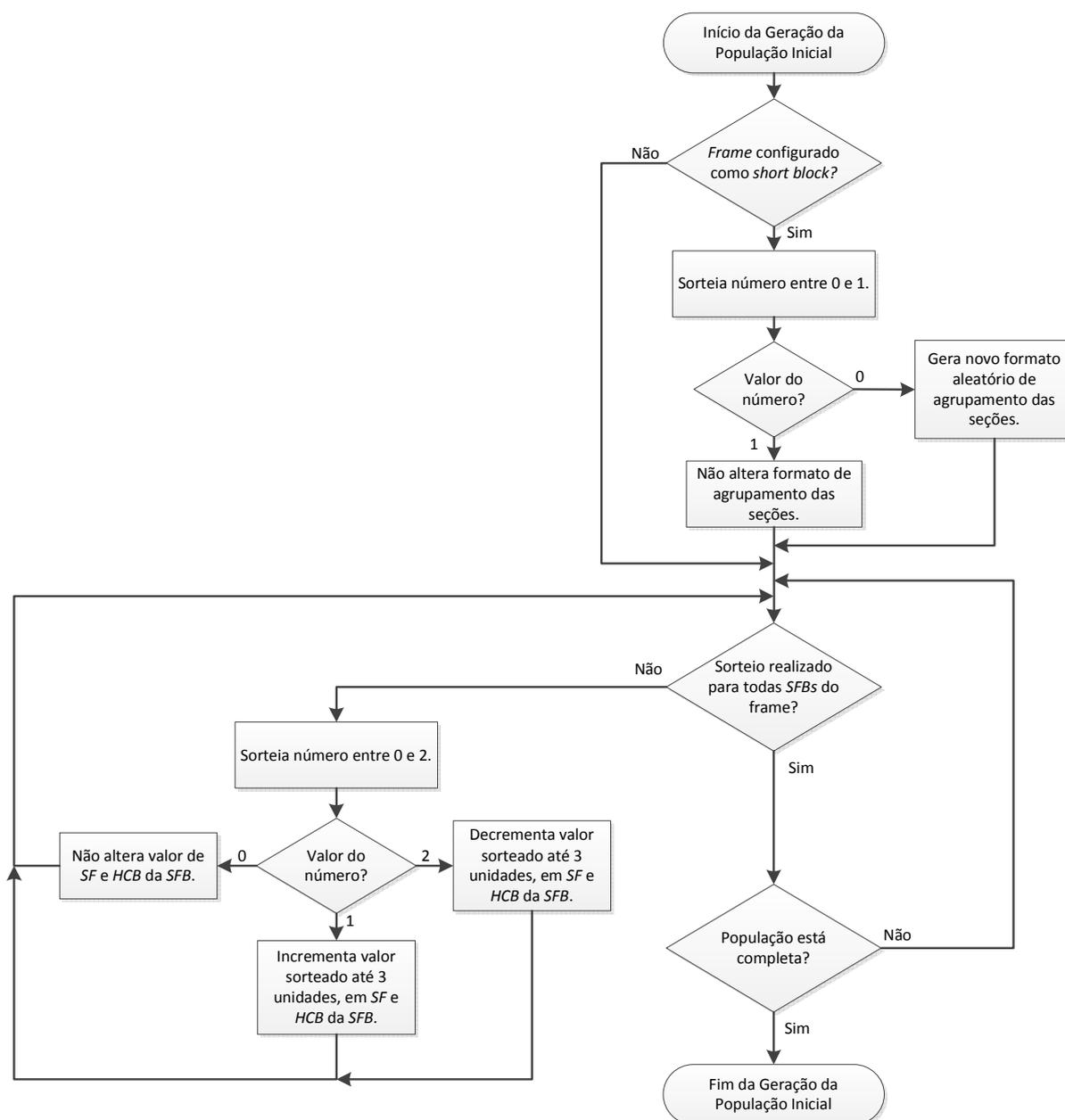


Figura 36 – Geração da população inicial para o Algoritmo Genético

As alterações inseridas sobre os valores encontrados pela *TLS* foram diferentes para o agrupamento dos blocos em seções, e para os valores de *scalefactors* e Códigos de Huffman. A solução da *TLS* para o agrupamento dos blocos foi alterada para uma configuração gerada aleatoriamente, com probabilidade de 50%. Assim a probabilidade de que o formato de agrupamento encontrado pela *TLS* fosse mantido, foi de 50%. Para os valores de *scalefactors* e Códigos de Huffman foi inserida uma variação de até três unidades, para mais ou para menos, sobre os valores da solução entregue pela *TLS*. Desta forma, para cada uma das *SFBs* de um *frame*, foi sorteada uma condição de alteração: incremento de até três unidades no valor de *SF* e *HCB*, decremento de até três unidades no valor de *SF* e *HCB*, ou a manutenção do valor original de *SF* e *HCB*.

É interessante observar que o tamanho da população foi definido como 20 indivíduos. Desta forma, este parâmetro não teve seu valor alterado durante os experimentos de parametrização apresentados no capítulo 7.

6.3.1.3 Operador de Seleção

No Algoritmo Genético implementado foi utilizado o processo de Seleção por Torneio. Desta forma, inicialmente todos os indivíduos foram ordenados do melhor para o pior, de acordo com sua aptidão para a FO. Depois, esta lista ordenada foi dividida ao meio, obtendo-se dois grupos, os dos mais aptos e os dos menos aptos. Assim, um indivíduo de cada grupo disputou a seleção para ser pai, através de um torneio baseado em um sorteio de um número entre 0 a 100. Para valores menores que 80, o indivíduo do grupo dos mais aptos venceu a disputa, e se tornou o escolhido. Porém, para valores superiores a 80, o indivíduo do grupo dos menos aptos venceu o torneio e foi selecionado para a etapa de cruzamento. Este processo de seleção privilegia a permanência das características dos indivíduos mais aptos na população da próxima geração, e desta forma, seu parâmetro de probabilidade foi configurado com 80%. É interessante observar que existe a probabilidade de 20% dos indivíduos pertencentes ao grupo dos menos aptos vencerem o torneio. Esta parcela dos menos aptos participarem do cruzamento contribui para manter a diversidade entre os indivíduos da nova geração.

6.3.1.4 Operadores de Cruzamento

Para a realização do processo de cruzamento dos indivíduos, foram criados métodos diferentes para cada uma das três variáveis de decisão do Problema de Codificação AAC abordadas neste trabalho. Isso se fez necessário, devido ao formato de representação de cada variável ser diferente dentro do cromossomo. Um mesmo método seria incapaz de realizar o cruzamento dos indivíduos que possuem vetores com diferentes conceitos de formatação. Desta forma, foram implementados dois métodos: o Método de Cruzamento 1, para as variáveis de decisão, *scalefactors* e *HCB*, e o Método de Cruzamento 2 para a variável de agrupamento dos *short blocks* em seções. A dinâmica dos cruzamentos é apresentada na Figura 37.

O processo de cruzamento é iniciado pela etapa de seleção de dois indivíduos. Na sequência é definido se os dois indivíduos selecionados serão cruzados, para que seus filhos possam ser incorporados à nova população, ou se eles próprios serão integrados à nova população, sem que o processo de cruzamento seja realizado. O parâmetro de percentual de cruzamento foi definido com probabilidade de 90%. No caso de realização do processo de cruzamento, o Método de Cruzamento 1 sempre é realizado, enquanto que o Método de Cruzamento 2, apenas se o *frame* for configurado como *short block*.

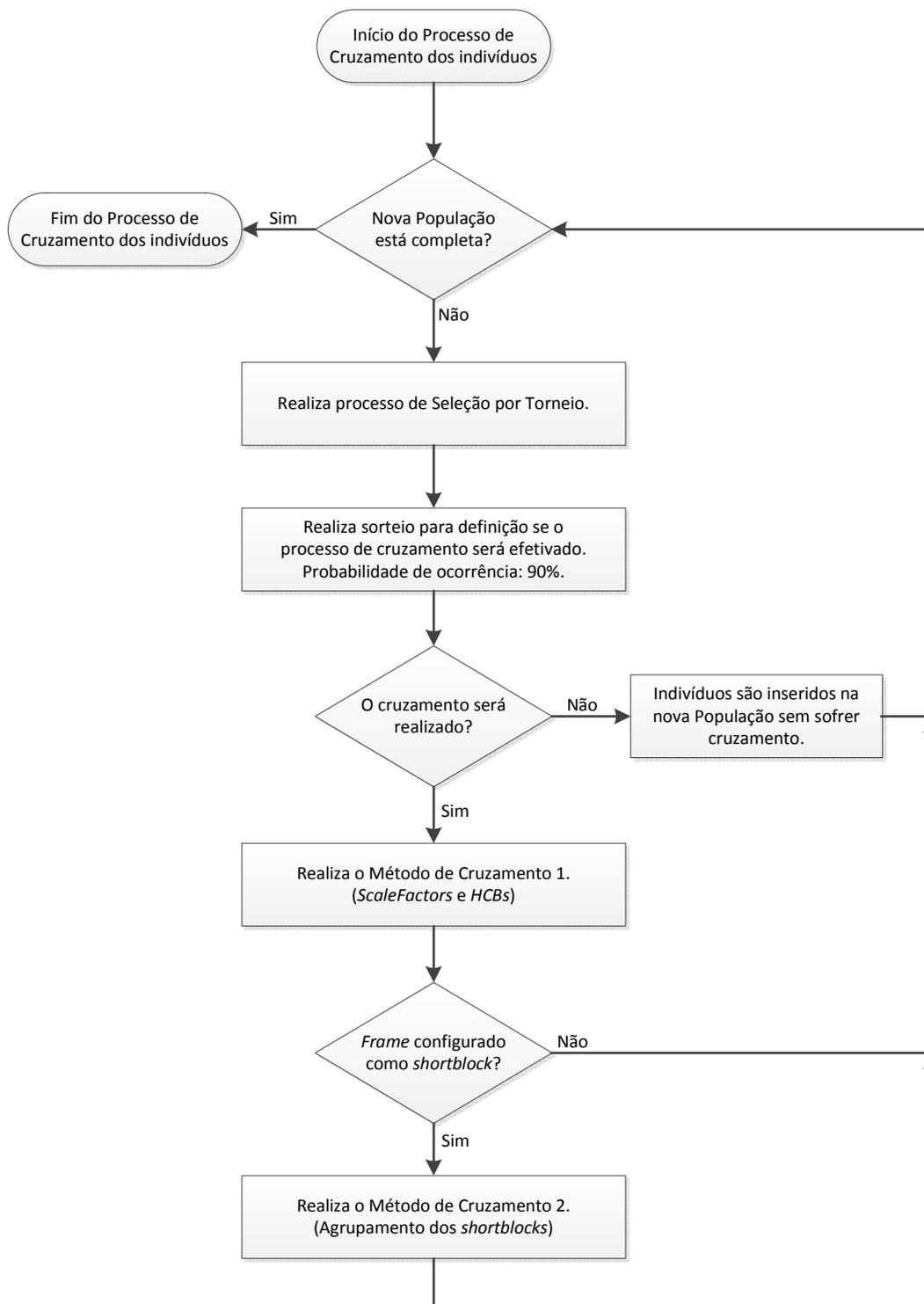


Figura 37 – Fluxograma do processo de cruzamento

O Método de Cruzamento 1 foi baseado na técnica do cruzamento aritmético combinada com o cruzamento de ponto único (descritas na seção 4.2.4), e foi aplicado para todas as *SFBs* do *frame*. Desta forma, foi possível obter novos valores

para as variáveis *SF* e *HCB*, mas também preservando parte das características genéticas dos pais. O diagrama do Método de Cruzamento 1 é apresentado no Figura 38.

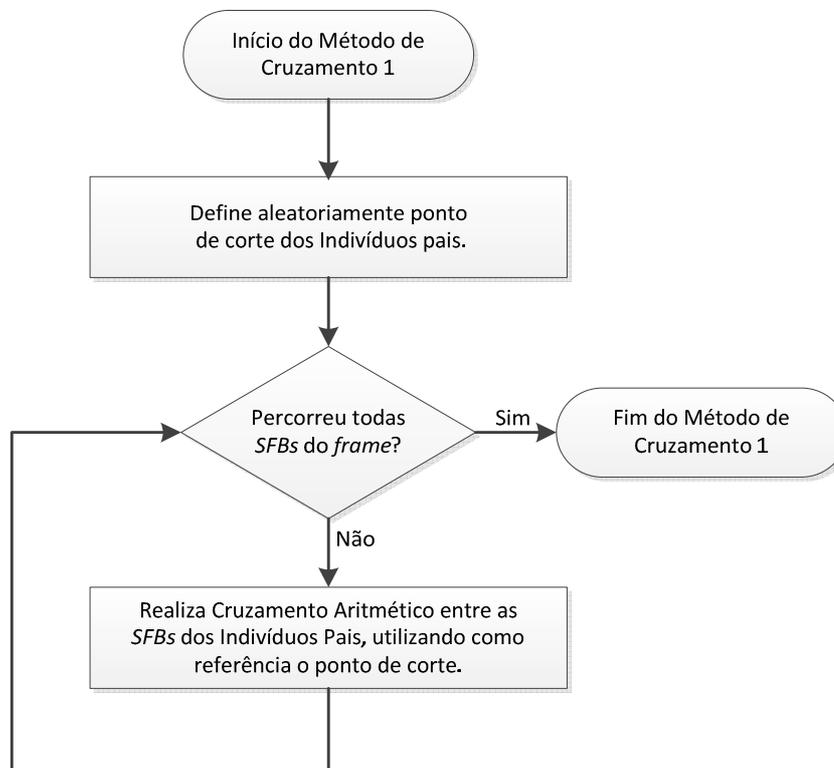


Figura 38 – Fluxograma do Método de Cruzamento 1

Deve-se observar que a quantidade de *SFBs* varia de acordo com a configuração do bloco definido pelo codificador, e isso implica diretamente em escolher, aleatoriamente, um ponto de corte válido para uma determinada configuração de *frame*.

O Método de Cruzamento 2 apenas realiza o cruzamento do agrupamento dos *short blocks* em seções. Este método inicia com um sorteio de uma seção do primeiro Pai, que será inserida em um filho sorteado, filho 1 ou 2. Este mesmo processo de sorteio, é realizado para uma seção do Pai 2. As seções são inseridas nos filhos 1 e 2, até que tenham no mínimo oito *short blocks*. É possível que através do processo de cruzamento, o limite de oito *short blocks* seja ultrapassado. Para contornar esta situação, sempre é realizada uma adequação dos filhos, para que

estes jamais tenham mais de oito *short blocks* em um *frame*. Este processo do Método de Cruzamento 2 é ilustrado na Figura 39.

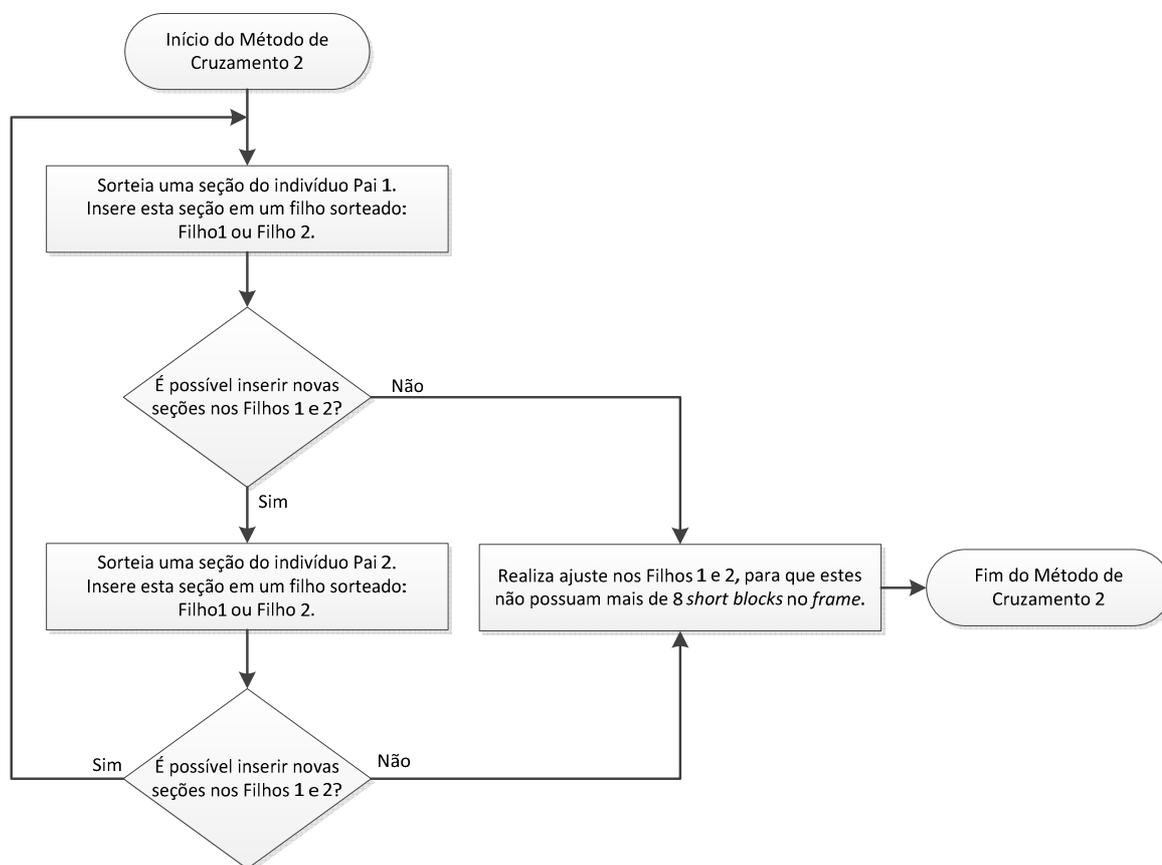


Figura 39 – Fluxograma do Método de Cruzamento 2

Para que seja possível visualizar a formação dos indivíduos através do Método de Cruzamento 2, é apresentada a sequência das Figura 40, Figura 41 e Figura 42.

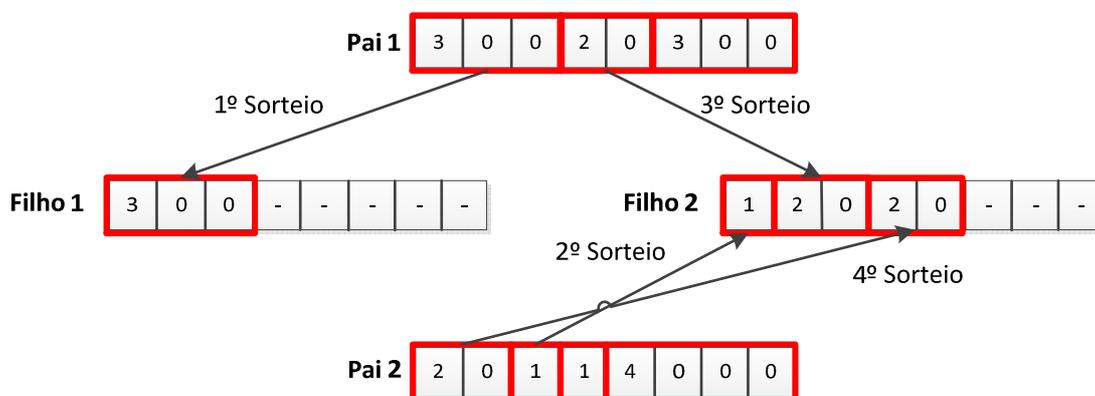


Figura 40 – Primeira rodada do Método de Cruzamento 2

Através da Figura 40, pode-se identificar as quatro primeiras seções sorteadas dos Pais 1 e 2, sendo distribuídas aleatoriamente nos Filhos 1 e 2. Na Figura 41 é apresentada a segunda rodada do Método de Cruzamento 2, na qual, novos quatro sorteios foram realizados. É interessante observar que no caso do exemplo, após oito sorteios, ambos os filhos foram completos pelas seções dos Pais, porém para garantir que se tornassem soluções viáveis para o Problema de Codificação AAC, estes foram submetidos à etapa de adequação. Os *short blocks* que são marcados na cor cinza, devem ser excluídos do *frame*, pois fazem a quantidade de blocos ultrapassar o valor máximo de oito.

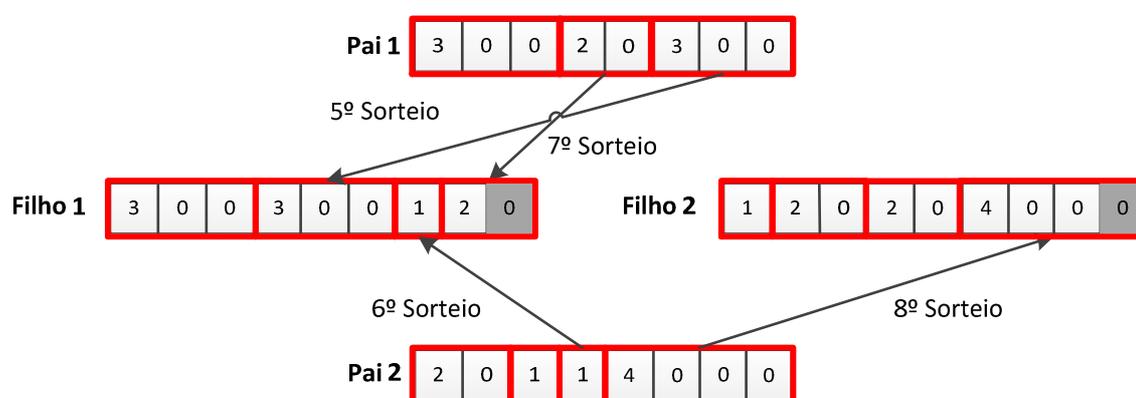


Figura 41 – Segunda rodada do Método de Cruzamento 2

Por fim, a Figura 42 apresenta os Filhos 1 e 2 gerados pelo Método de Cruzamento 2, após a etapa de adequação das soluções.

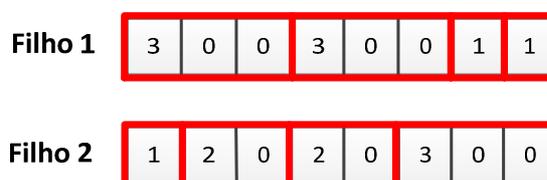


Figura 42 – Filhos resultantes do Método de Cruzamento 2

Após a adequação da quantidade de blocos do *frame*, os indivíduos filhos passam a representar uma parte da solução do Problema de Codificação AAC, e desta forma, são inseridos na população da nova geração.

6.3.1.5 Operadores de Mutação

De maneira análoga ao processo de cruzamento, devido às diferenças no formato da codificação do cromossomo das variáveis de decisão, foi necessária a criação de dois métodos de mutação: o Método de Mutação 1 e o Método de Mutação 2. O Método de Mutação 1 foi aplicado às variáveis de decisão, *scalefactors* e *HCB*. Já a variável de agrupamento dos *short blocks* em seções, foi submetida ao Método de Mutação 2. O fluxograma do processo de mutação é apresentado na Figura 43.

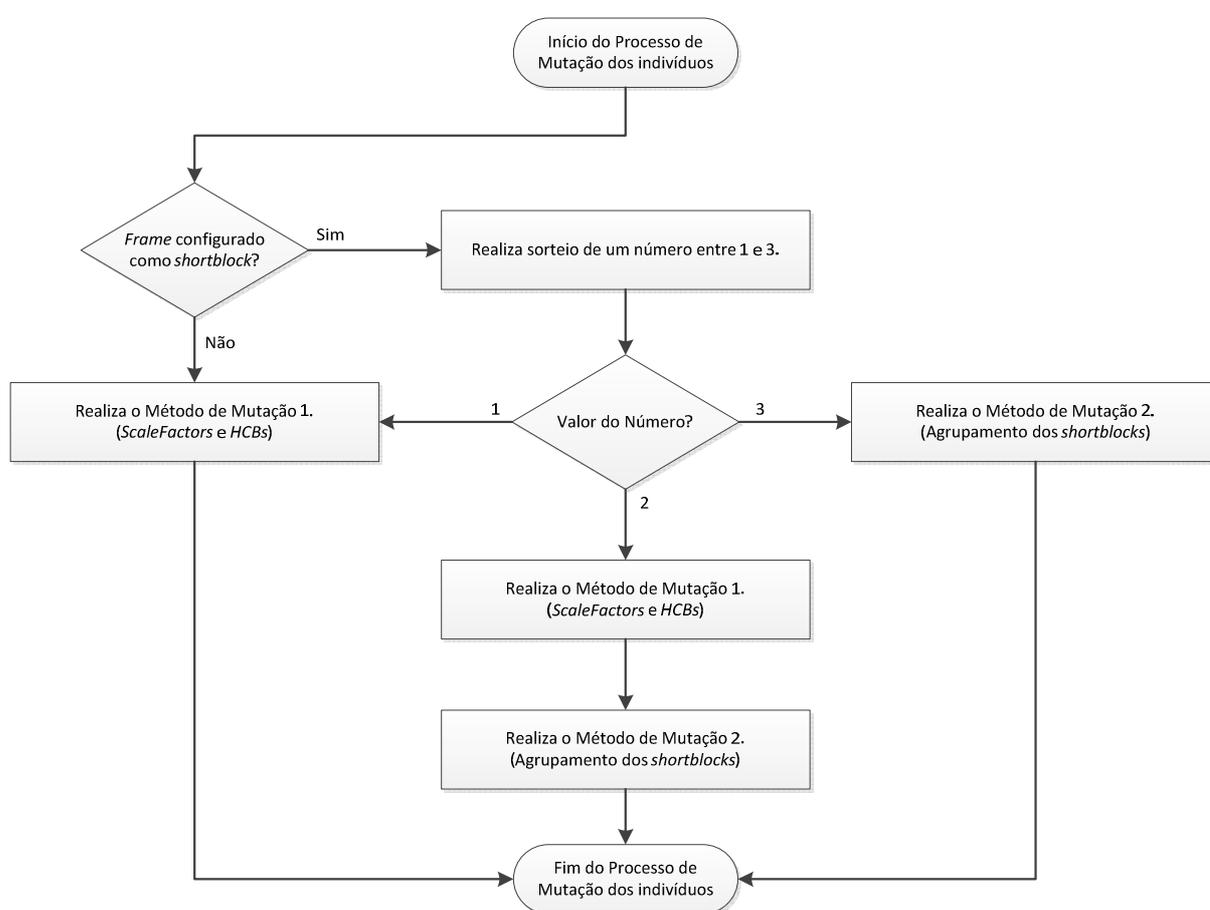


Figura 43 – Fluxograma de Mutação do Algoritmo Genético

O Método de Mutação 1 atua sobre os valores dos *scalefactors* e *HCB*, sorteando as *SFBs* que terão seus valores alterados. Diferentemente dos valores de *HCB*, os valores dos *scalefactors* não podem ser alterados sem controle, pois senão as soluções se tornarão inviáveis por não atender a uma das restrições do

codificador AAC. Como apresentado na seção 6.2, os *scalefactors* são codificados diferencialmente, e desta forma, a diferença entre dois *scalefactors* adjacentes não pode ser maior que 60 (ISO/IEC, 2004). Na Figura 44 é apresentado o fluxograma do Método de Mutação 1.

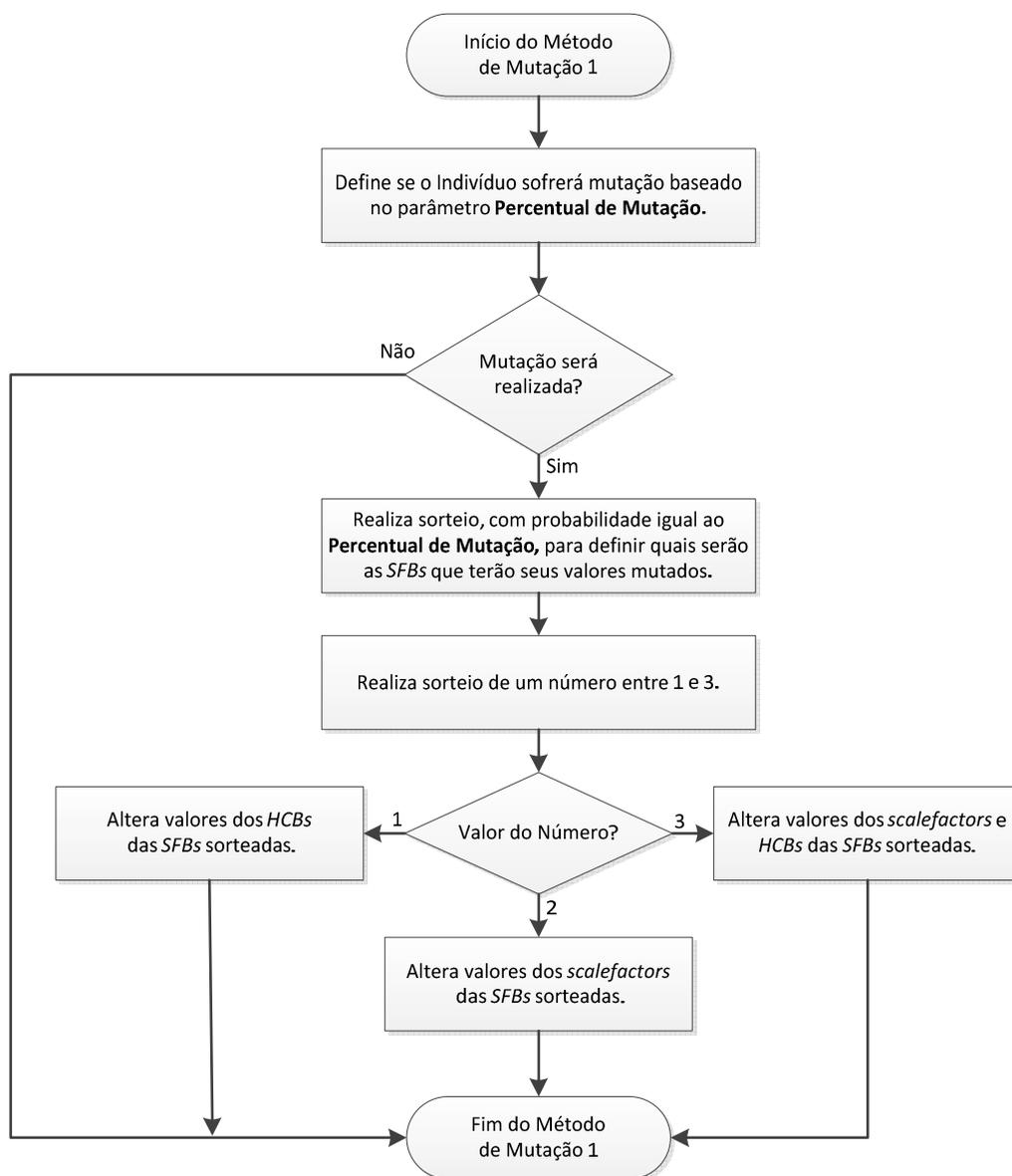


Figura 44 – Fluxograma do Método de Mutação 1

Como visto na Figura 43, o Método de Mutação 2 somente pode ser realizado, se o *frame* for configurado como *short block*. Uma vez que esta restrição é atendida, a ocorrência do Método de Mutação 2 ainda é condicionada à probabilidade definida pelo parâmetro Percentual de Mutação. O processo

executado neste método provoca a edição, criação ou eliminação das seções do *frame* de uma determinada solução que sofrerá mutação. O fluxograma deste sistema é apresentado na Figura 45.

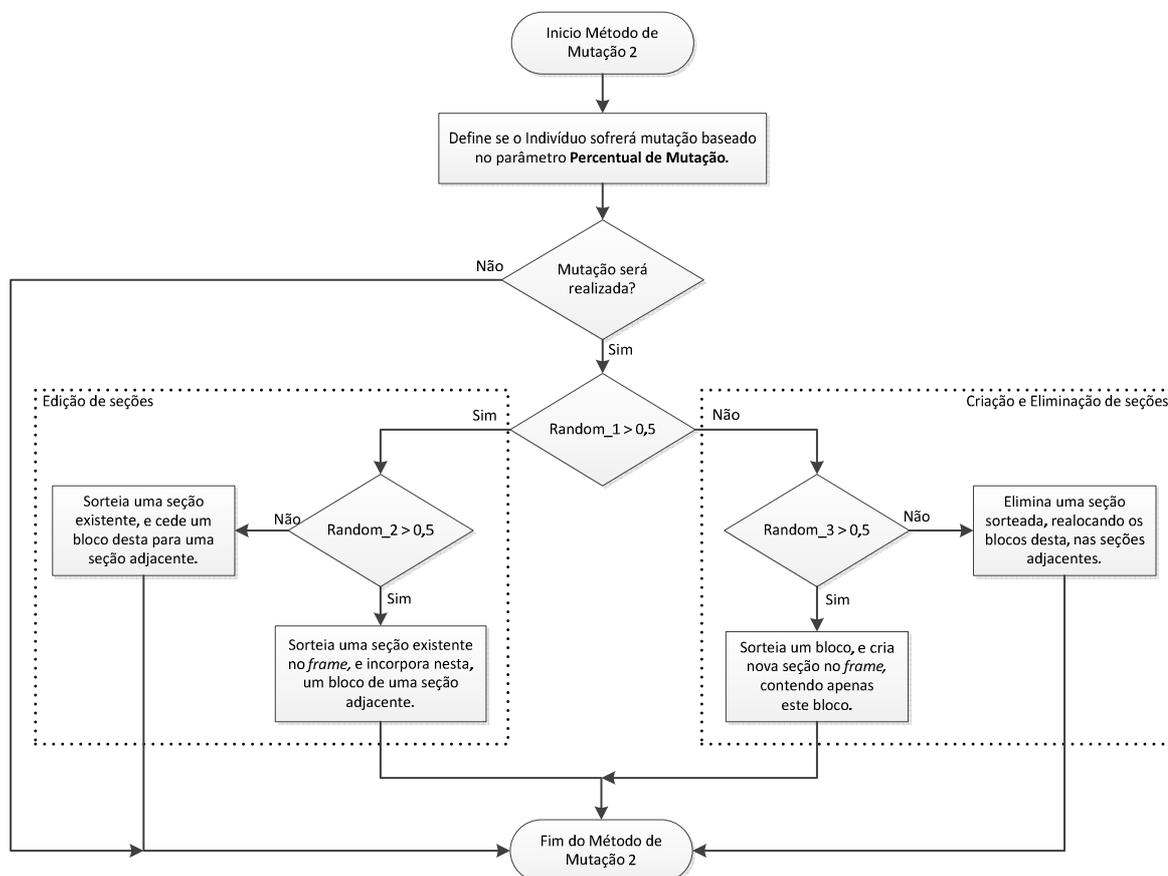


Figura 45 – Fluxograma do Método de Mutação 2

6.3.1.6 Critério de Parada

Para que o Algoritmo Genético seja finalizado é necessário um critério que determine o seu encerramento. Na implementação deste trabalho, este critério foi definido como sendo o Número de Gerações. Isso significa que o Algoritmo Genético é encerrado quando atingir o Número de Gerações pré-definido. Este é um dos parâmetros das metaheurísticas que foram alterados durante a realização dos experimentos de parametrização descritos no capítulo 7.

6.3.2 Algoritmo Busca Tabu para o Problema de Codificação AAC

Nesta seção é apresentado o desenvolvimento e implementação do algoritmo da metaheurística Busca Tabu aplicado ao Problema de Codificação AAC. São descritos o processo de codificação da solução, a obtenção da solução inicial, a geração das estruturas de vizinhança, a avaliação da função objetivo, a determinação e formação da Lista Tabu, e definição do critério de aspiração e critério de parada.

6.3.2.1 Codificação da Solução

A codificação da solução utilizada na implementação do algoritmo da Busca Tabu, faz o uso da mesma formatação da codificação do cromossomo do Algoritmo Genético, apresentada na seção 6.3.1.1.

6.3.2.2 Solução Inicial

A solução inicial utilizada pela Busca Tabu foi gerada pelo Algoritmo Genético. Desta forma, a melhor solução encontrada até o término da execução do Algoritmo Genético, é assumida como solução inicial do algoritmo Busca Tabu.

6.3.2.3 Estruturas de Vizinhança

De maneira análoga ao Algoritmo Genético, devido à diferença do formato de codificação das variáveis de decisão em estudo, foram desenvolvidos diferentes métodos de geração de vizinhança a partir da solução atual. Desta forma foram denominados: Método de Geração 1 para alteração dos valores dos *scalefactors* das *SFBs*, Método de Geração 2 para alteração dos valores dos *HCBs* das *SFBs*, e Método de Geração 3 para alteração do agrupamento das seções do *frame*.

A dinâmica do sistema de geração da vizinhança da Busca Tabu pode ser visualizada na Figura 46.

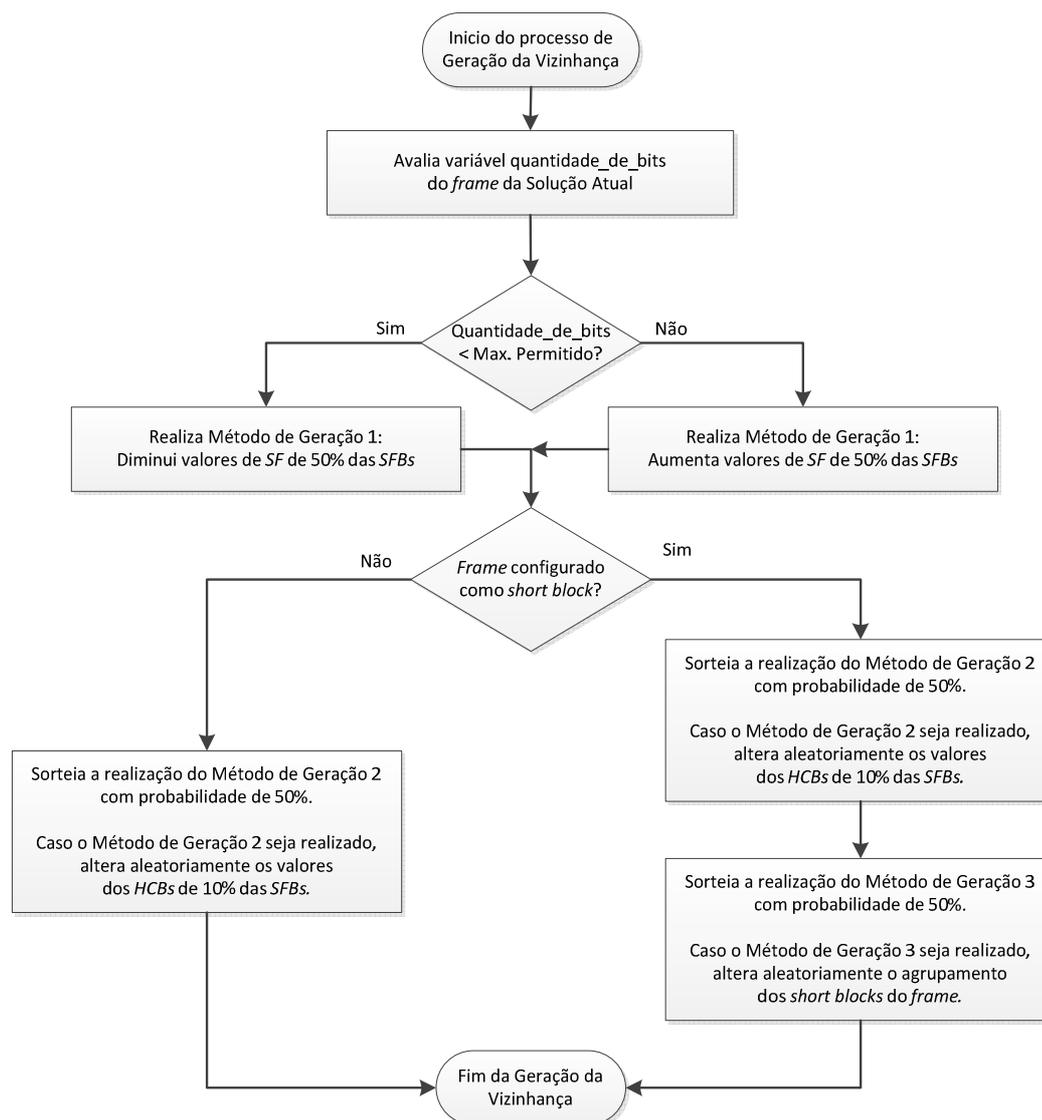


Figura 46 – Fluxograma de geração de vizinhança

Inicialmente, é avaliada a quantidade de bits que é utilizada na solução atual. Caso a quantidade de bits seja inferior ao limite máximo de bits por *frame*, então os valores de *scalefactors* são diminuídos, de maneira que são necessários mais bits para realizar a codificação. Porém se quantidade de bits é maior do que a máxima permitida em um *frame*, então os *scalefactors* têm seus valores aumentados, de forma a diminuir a quantidade de bits necessária para a codificação do *frame*. Assim, deve-se observar que o Método de Geração 1 é alterado de acordo com a quantidade de bits necessários para codificar o *frame* da solução atual. Este processo de alteração dos valores, conforme proposto pelo Método de Geração 1, é utilizado para conduzir a busca por soluções em regiões do espaço de soluções em

que a quantidade de bits do processo de codificação seja utilizada em sua totalidade, e assim possa garantir uma qualidade perceptual do áudio codificado. Devido às outras duas variáveis de decisão não impactarem com tanta intensidade, sobre a qualidade perceptual do sinal de áudio codificado, os Métodos de Geração 2 e 3 são utilizados de maneira igual, independente da quantidade de bits do *frame* atual.

Os Métodos de Geração de vizinhança são detalhados abaixo:

- **Método de Geração 1:**

Método de geração de vizinhança que altera apenas os valores dos *scalefactors* das *SFBs*. Nem todas as *SFBs* de um *frame* têm seus *scalefactors* alterados. A probabilidade de que as *SFBs* tenham os valores de seus *scalefactors* alterados é de 50%. É interessante observar que o processo de alteração implica em aumentar ou diminuir os valores dos *SFs*, atendendo às restrições do modelo computacional, expostas na seção 6.2, em função da quantidade de bits necessários para codificar o *frame*.

- **Método de Geração 2:**

De maneira análoga ao Método de Geração 1, este método apenas altera os valores dos *HCBs* das *SFBs*. Buscando a intensificação da busca, a probabilidade de que uma *SFB* tenha o seu valor de *HCB* alterado é de apenas 10%. Desta forma, nem todas as *SFBs* de um *frame* tem seus valores de *HCB* alterados por este método. A alteração é aplicada independente do valor da quantidade de bits do *frame*.

- **Método de Geração 3:**

Este método, diferente dos demais, não é aplicado sobre as *SFBs*, mas sobre o agrupamento das seções do *frame*. Deve-se observar que este método de geração de vizinhança somente é executado se o *frame* for configurado como *short block*, devido à possibilidade de agrupamentos dos 8 *short blocks* de um *frame* em seções, conforme descrito na seção 3.2.5.

Na Figura 47 é apresentado o fluxograma do Método de Geração 3 de vizinhança. Assim, inicialmente é decidido aleatoriamente, se o processo de geração será baseado em uma edição das seções existentes no *frame*, ou

baseado na criação ou eliminação de seções no *frame*. Caso seja escolhido o processo de edição das seções, este é realizado sobre uma seção sorteada do *frame*, de maneira que esta pode vir a ter a sua seção incrementada ou decrementada em um bloco de uma seção adjacente. Caso o processo de criação ou eliminação das seções seja escolhido, então é sorteado um dos oito blocos do *frame*. No processo de criação, este bloco sorteado formará uma nova seção, e os blocos que pertenciam à mesma seção deste, permanecem formando suas seções. No caso do processo de eliminação, a seção que possui o bloco sorteado deixa de existir, e os blocos pertencentes a esta, são realocados nas seções adjacentes por sorteio.



Figura 47 – Fluxograma do Método de Geração 3 de Vizinhança

O Método de Geração 3 de vizinhança, se assemelha com o Método de Mutação 2 do Algoritmo Genético, aplicado sobre a variável de agrupamento de blocos. Deve-se observar também, que apenas umas das modificações citadas (edição, criação ou eliminação) é executada por vez sobre as seções do *frame*.

6.3.2.4 Função Objetivo

Através da Função Objetivo é calculado o valor de *ANMR* para cada *frame* de cada uma das soluções candidatas. Desta forma, é escolhida da vizinhança a solução que apresentar menor valor de *ANMR*. Neste ponto é muito importante observar, que além da solução possuir o menor valor de *ANMR*, esta não deve ultrapassar a quantidade de bits máxima estimada para um *frame*. Desta forma, soluções que contabilizem uma quantidade de bits maior do que a permitida para um *frame* são descartadas.

6.3.2.5 Lista Tabu

O parâmetro que deve ser inserido na Lista Tabu para representar os movimentos realizados durante a busca, deve ser capaz de representar uma determinada combinação das variáveis de decisão, de maneira que esta condição possa referenciar ao algoritmo um estado da busca. No Problema de Codificação de AAC, o parâmetro que foi utilizado para definir um determinado movimento da busca, foi a quantidade de bits necessários para codificar o *frame*. O valor da quantidade de bits de um *frame*, é obtido justamente em função das variáveis de decisão, e representa uma determinada combinação entre elas. Desta forma, através deste parâmetro, o algoritmo da Busca Tabu pôde evitar que sua busca em trajetória retornasse tão cedo a regiões do espaço de soluções já visitadas.

6.3.2.6 Critério de Aspiração

O Critério de Aspiração utilizado no algoritmo Busca Tabu implementado, permite que a situação de Tabu de um determinado movimento, apenas seja perdoada, caso a solução obtida pelo movimento Tabu seja a melhor de todas as soluções encontradas até o momento. Porém, para que a situação Tabu seja efetivamente perdoada, foi estipulada a necessidade de que esta solução Tabu permanecesse na Lista Tabu durante um determinado período. Desta forma, foi utilizado um período mínimo de permanência da solução Tabu na Lista, de $2/3$ do seu tamanho total. Assim, ao restar apenas $1/3$ do seu tamanho total para que a

solução Tabu seja eliminada da Lista, esta poderá ser perdoada, caso a busca não tenha encontrado outra solução melhor que esta.

6.3.2.7 Critério de Parada

O Critério de Parada utilizado na implementação do algoritmo Busca Tabu foi definido pelo número de iterações sem melhora da Função Objetivo (BT_Max). Este parâmetro foi alterado nos experimentos de parametrização apresentados no capítulo 7.

6.3.3 Algoritmo Híbrido

A estrutura de simulação para otimização do codificador MPEG-4 AAC, proposta pelo trabalho é baseada em uma solução híbrida das metaheurísticas Algoritmo Genético e Busca Tabu.

Uma solução híbrida das metaheurísticas AG e BT tem apresentado através de resultados experimentais, melhores soluções quando comparadas às soluções dos algoritmos em separado (LAN-YING e MIN, 2008). A ideia é que em uma única estrutura, as duas metaheurísticas em conjunto possam compartilhar suas melhores características, de maneira que consigam diversificar e intensificar a busca pelo espaço de soluções (GLOVER, KELLY e LAGUNA, 1995). Desta forma, dentro da estrutura híbrida implementada, inicialmente é executada a metaheurística AG para que as soluções (indivíduos) sejam distribuídas pelo espaço de soluções. Assim, através da diversificação, o AG procura identificar regiões promissoras, e expandir a busca para regiões do espaço de soluções que ainda não foram exploradas (GENDREAU, LAPORTE e POTVIN, 2001). Para contrapor este fato é necessário intensificar a busca, e é por isso que na estrutura de otimização híbrida é implementada a Busca Tabu na sequência do Algoritmo Genético. Como a BT é um método baseado em Busca Local, o processo de busca dispensa um bom tempo em uma parte restrita do espaço de soluções, intensificando a busca nas áreas promissoras identificadas pelo AG.

A Figura 48 apresenta o esquema de como as duas metaheurísticas em conjunto resolvem o Problema de Codificação AAC para cada *frame* do sinal de áudio.

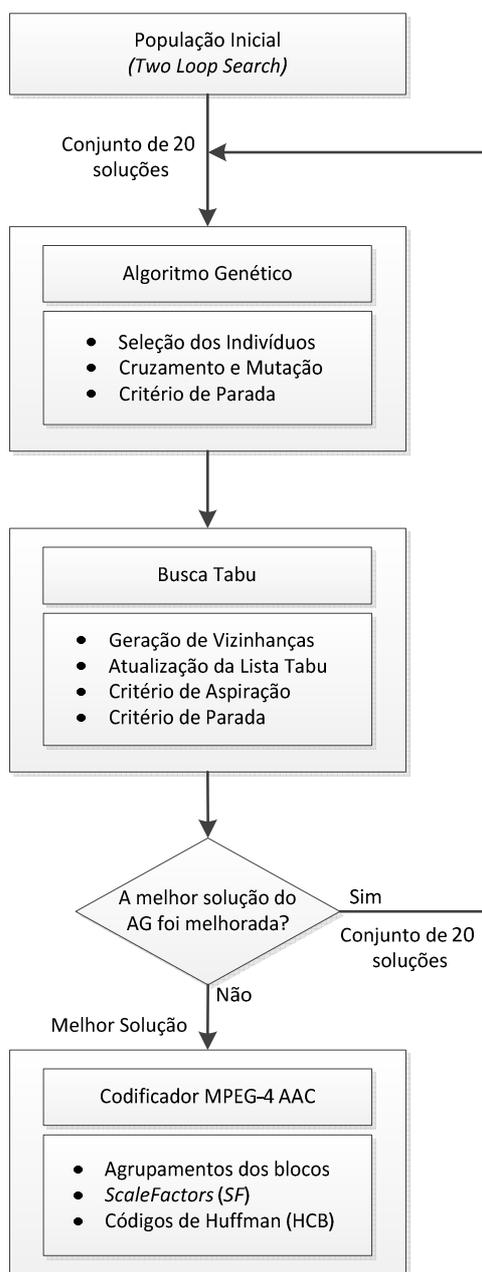


Figura 48 – Estrutura de otimização híbrida para o modelo computacional

Para facilitar a observação da dinâmica dos acontecimentos do diagrama é interessante uma análise do sistema dividido em etapas:

- **Etapa 1:**

Inicialmente, para que a metaheurística AG possa ter um ponto de partida é necessário que se obtenha uma População Inicial. Este conjunto de soluções iniciais é gerado pelo processo de iteração já existente (*TLS*) dentro do codificador MPEG-4 AAC de referência. Assim, após uma execução do Laço Externo e uma execução do Laço Interno (seção 3.2.6), obtém-se uma solução. Devido esta ser uma solução única de um método exato, para compor uma população com diferentes indivíduos, a solução é ainda aleatoriamente modificada conforme o diagrama da Figura 36.

- **Etapa 2:**

Através da População Inicial, o AG gera novos indivíduos através do processo de cruzamento, e diversifica-os através da mutação. Após este processo, então é selecionado da população, o melhor indivíduo, aquele que obteve o menor valor para a Função Objetivo (*ANMR*). Este indivíduo, então é aplicado à entrada da BT, de maneira que esta solução representada pelo melhor indivíduo, é assumida como a nova solução inicial da BT.

- **Etapa 3:**

De posse da solução inicial, a BT gera soluções vizinhas, intensificando a procura e criando uma busca em trajetória pelo espaço de soluções. As melhores soluções obtidas durante a trajetória de busca são armazenadas em uma nova população, que poderá ser aplicada novamente à entrada do Algoritmo Genético. A Busca Tabu é executada até que seu critério de parada, definido por um número determinado de iterações, seja atendido. Caso a melhor solução encontrada pela BT, seja melhor que sua solução inicial, então a nova população formada durante a trajetória da Busca Tabu, é entregue novamente para o Algoritmo Genético.

- **Etapa 4:**

Este processo se repete até que a Busca Tabu não obtenha sucesso em melhorar a solução entregue pelo Algoritmo Genético. Neste caso, o Algoritmo Híbrido é encerrado e espera-se que uma boa solução para o Problema de Codificação AAC tenha sido encontrada.

Com o encerramento do Algoritmo Híbrido, a solução encontrada pela estrutura híbrida de otimização, é repassada para o codificador MPEG-4 AAC, para que ele, a partir dos dados encontrados pelas metaheurísticas, possa realizar a codificação do *frame* em análise. Os dados do *ANMR* para cada *frame* foram arquivados, para que se pudesse avaliar a qualidade perceptual do sinal de áudio codificado com os parâmetros obtidos pelas metaheurísticas. Desta forma, foi possível efetuar uma comparação entre a qualidade perceptual obtida com a estrutura de referência *TLS* do codificador, e a estrutura de otimização de metaheurísticas desenvolvida neste trabalho.

7 VALIDAÇÃO E EXPERIMENTOS

Neste capítulo são apresentados os resultados obtidos com o processo de validação do modelo computacional implementado neste trabalho, os experimentos realizados na etapa de parametrização das metaheurísticas, e também os resultados finais com os experimentos de aplicação.

7.1 Biblioteca de Arquivos de Áudio Teste

Para que fosse possível avaliar a estrutura de simulação do codificador desenvolvida para resolver o Problema de Codificação AAC, foi necessário efetivamente realizar a codificação de áudio. Como o codificador de áudio e a estrutura de otimização foram programados utilizando a linguagem C ANSI, os experimentos e testes de validação, foram realizados através de arquivos de áudio. Para isso foram utilizados arquivos de áudio no formato WAV como entrada do codificador, que gerou em sua saída arquivos de áudio comprimidos na extensão AAC.

O formato WAV para arquivos de áudio foi criado através de uma parceria entre IBM e Microsoft, com o intuito de padronizar uma forma de armazenamento de áudio nos computadores. A técnica consiste em uma codificação PCM do sinal de áudio, cujo processo é caracterizado por uma baixa compressão (MOCK, 2004). Os arquivos WAV possuem a mesma qualidade perceptual encontrada nos CDs, com uma taxa aproximadamente 10 megabytes por minuto de áudio. Devido a estes fatores, e também a facilidade de acesso e manipulação do formato, os arquivos WAV são utilizados como fonte de sinais para os diversos codificadores de áudio (MOCK, 2004). Desta forma, na abordagem deste trabalho os arquivos WAV foram utilizados como entrada do codificador para geração de novos arquivos codificados em AAC.

Para a realização dos experimentos de codificação é necessário um conjunto de arquivos de áudio teste, contemplando diferentes tipos de música e sons. Estes arquivos de teste se fazem necessários, para que se possa analisar a qualidade perceptual obtida com o codificador, com os diferentes tipos de sons em sua entrada. Sons mais complexos, como por exemplo, uma orquestra ou uma banda de

rock, possuem uma alta entropia musical (TANKANOW, 2007), pois diversos elementos ocorrem juntos no tempo com pouca redundância de informação, e por isso apresentam maior dificuldade ao codificador no processo de compressão. Já os sons que basicamente envolvem sinais de áudio compostos apenas por vozes como de um telejornal, por exemplo, são mais simples (TANKANOW, 2007). Como os sinais de vozes possuem uma faixa de frequência restrita, e especificamente em um telejornal, não são somados a outras diversas fontes de som, sua entropia é baixa, e assim são classificados como sons simples (SINGH, 2011). Desta forma, baseado nos trabalhos (DERRIEN, DUHAMEL e CHARBIT, 2006), (BAUER e VINTON, 2006), (TANKANOW, 2007), (WU e QIU, 2008) e (SINGH, 2011), foi elaborada uma biblioteca própria de arquivos de áudio teste, para ser utilizada na realização dos experimentos deste trabalho. A lista dos sons que compõem a biblioteca de áudio teste é apresentada na Tabela 8.

Índice	Tipo de Som	Descrição	Tempo
1	Bateria	<i>DG Samples - Rock Beat Drums 02</i>	8.30s
2	Blues	<i>All Blues - Kora Jazz Band</i>	15.30s
3	Contrabaixo	<i>The Clairvoyant - Iron Maiden</i>	9.8s
4	Coral	<i>Symphony Nº 9 - Beethoven</i>	10.25s
5	<i>Crash China</i>	<i>DG Samples - Crash China</i>	3.25s
6	Cravo	<i>Oeuvres Pour Clavecin - François Couperin</i>	10.2s
7	Narração	<i>DG Samples - Vocals Shout 120</i>	10.80s
8	Orquestra	<i>Symphony Nº 9 - Beethoven</i>	15.70s
9	Rock	<i>It's my Life - Bon Jovi</i>	15.15s
10	Sax	<i>DG Samples - Sax Riff 128 C</i>	8.25s

Tabela 8 – Arquivos WAV utilizados nos experimentos

Todos os arquivos WAV listados possuem um canal de áudio, caracterizando um sinal mono. Também se deve observar, que todos os sons dos arquivos foram amostrados com frequência de amostragem de 44100Hz, sendo cada amostra representada por 16 bits.

7.2 Validação

A finalidade do processo de validação é verificar o comportamento do modelo computacional do codificador MPEG-4 AAC implementado, e confrontar com a dinâmica real do mesmo, descrita na documentação da norma (ISO/IEC, 2004) (ISO/IEC, 2005) e literatura auxiliar. Desta forma, foram realizados experimentos com o modelo computacional, a fim de identificar de que maneira as três variáveis de decisão abordadas neste trabalho, impactam sobre a métrica de qualidade perceptual *ANMR* e a taxa de bits. Estas duas métricas foram escolhidas, pois a qualidade perceptual é a métrica representada pela Função Objetivo, e a taxa de bits é a principal restrição modelada para o Problema de Codificação AAC.

O objetivo dos experimentos de validação foi estudar o comportamento individual de cada uma das variáveis de decisão. Desta maneira, durante o experimento, a variável de estudo teve seus valores alterados, enquanto que as outras duas foram mantidas com seus valores constantes, totalizando três experimentos distintos de validação. Deve-se observar que a taxa de bits não foi fixada durante os experimentos de validação, justamente para avaliar seu comportamento em função das variáveis de decisão do Problema de Codificação AAC.

No primeiro experimento, para identificar o comportamento individual dos *scalefactors*, todos os *frames* dos sinais de áudio teste foram intencionalmente codificados como *long blocks*, e o quinto código de Huffman foi utilizado na configuração para todas as *SFBs* do *frame*. Desta maneira, ambas variáveis deixaram de influenciar na medida da qualidade perceptual e na taxa total de bits. Assim os valores de *scalefactors* puderam ser alterados dentro de sua faixa específica (ISO/IEC, 2004) (ISO/IEC, 2005), e a sua relação com a taxa de bits e a qualidade perceptual foi identificada. Desta forma, os dez arquivos de áudio teste foram codificados, e os dados da qualidade perceptual e taxa de bits foram arquivados para que a dinâmica das variáveis de decisão pudesse ser avaliada.

No modelo computacional a qualidade perceptual foi representada pela métrica *ANMR*, que é uma medida de distorção obtida através do cálculo de um somatório do erro de quantização, equação (24). Devido à faixa de valores desta métrica ser muito extensa, para facilitar a representação gráfica, estes valores foram

calculados em dB, tendo sempre como referência na relação de distorção, o valor da distorção obtida com o *scalefactor* igual a 60. Na Tabela 9 são apresentados os valores de distorção, em sua forma absoluta e em dB, obtidos para os dez arquivos de áudio teste, variando os valores dos *scalefactors* de 60 até 120.

Índice do Arquivo	Distorção ANMR	Valores dos <i>scalefactors</i>						
		60	70	80	90	100	110	120
1	Erro Quant.	41	398	3.404	13.910	53.000	146.971	175.502
	Relação dB	0,00	19,74	38,38	50,61	62,23	71,09	72,63
2	Erro Quant.	76	923	8.855	67.624	382.390	826.781	831.499
	Relação dB	0,00	21,69	41,33	58,99	74,03	80,73	80,78
3	Erro Quant.	28	337	3.589	28.912	213.283	531.661	531.661
	Relação dB	0,00	21,61	42,16	60,28	77,64	85,57	85,57
4	Erro Quant.	55	659	6.888	59.243	269.144	410.862	410.862
	Relação dB	0,00	21,57	41,95	60,65	73,79	77,47	77,47
5	Erro Quant.	46	565	5.958	42.510	103.912	113.472	113.472
	Relação dB	0,00	21,79	42,25	59,31	67,08	67,84	67,84
6	Erro Quant.	61	721	6.753	44.671	135.025	155.029	155.029
	Relação dB	0,00	21,45	40,88	57,29	66,90	68,10	68,10
7	Erro Quant.	22	211	1.715	9.769	18.284	18.284	18.284
	Relação dB	0,00	19,64	37,84	52,95	58,39	58,39	58,39
8	Erro Quant.	44	505	5.237	40.612	183.341	230.612	230.612
	Relação dB	0,00	21,20	41,51	59,30	72,40	74,39	74,39
9	Erro Quant.	128	1.625	18.663	165.024	718.225	1.184.797	1.184.797
	Relação dB	0,00	22,07	43,28	62,21	74,98	79,33	79,33
10	Erro Quant.	43	495	4.900	37.897	229.795	315.204	315.204
	Relação dB	0,00	21,22	41,13	58,90	74,56	77,30	77,30

Tabela 9 – Análise dos *scalefactors* vs ANMR

Através da análise da Tabela 9 pode-se identificar para os dez arquivos de áudio teste, que com o aumento dos valores de *scalefactors*, a quantidade de distorção sobre o sinal de áudio codificado também aumenta. Este comportamento era esperado conforme descrito na norma do codificador MPEG-4 AAC (ISO/IEC, 2004) (ISO/IEC, 2005).

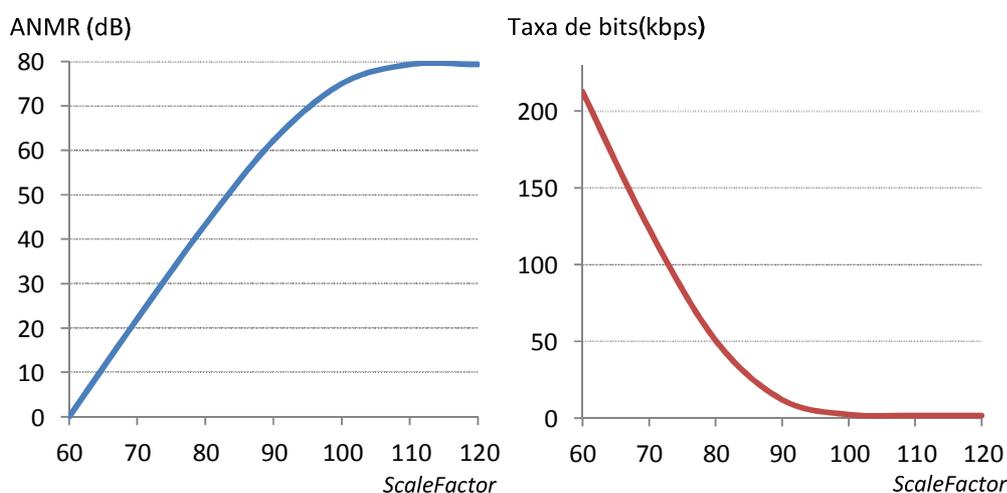
De maneira análoga, na Tabela 10, a taxa de bits em bps, foi obtida para cada um dos dez arquivos, ao variar dos valores dos *scalefactors* de 60 a 120.

Taxa de bits (bps)							
Índice do Arquivo	Valores dos <i>scalefactors</i>						
	60	70	80	90	100	110	120
1	117.812	41.277	9.636	1.745	1.281	1.215	1.211
2	136.422	65.255	21.520	6.040	2.129	1.647	1.646
3	47.304	23.339	8.766	3.473	1.813	1.394	1.394
4	105.635	52.277	20.772	5.687	2.000	1.792	1.792
5	96.175	49.920	17.634	3.822	1.762	1.722	1.722
6	135.298	65.780	21.035	4.208	1.701	1.655	1.655
7	46.624	15.230	4.204	1.444	1.136	1.136	1.136
8	89.787	41.624	15.539	4.535	1.790	1.681	1.681
9	212.586	123.266	50.660	11.984	2.338	1.651	1.651
10	87.939	40.472	15.342	5.179	2.049	1.738	1.738

Tabela 10 – Análise dos *scalefactors* vs taxa de bits

Pode-se verificar na Tabela 10, que um aumento nos valores de *scalefactors*, provoca uma diminuição na taxa total de bits. Este comportamento está diretamente relacionado aos dados da Tabela 9, que demonstram que para uma maior taxa de bits, menor é a distorção sobre o sinal de áudio codificado, e assim maior a qualidade perceptual.

Para ilustrar a influência dos *scalefactors* sobre a qualidade perceptual e taxa de bits, são apresentadas na Figura 49 as curvas de comportamento obtidas nos experimentos das Tabela 9 e Tabela 10, com o arquivo de áudio teste de número 9.

Figura 49 – Comportamento dos *scalefactors* vs ANMR e taxa de bits

A análise gráfica permite identificar que com a diminuição do valor do *scalefactor*, a distorção inserida no sinal de áudio também é diminuída, ou seja, sua qualidade perceptual é aumentada. Com a taxa de bits, a relação é contrária, quanto menor o valor do *scalefactor* maior a taxa bits. Estas duas relações demonstram a dinâmica do codificador como realmente ele deveria ser. Para menores valores de *scalefactors*, menor a distorção, e para que se obtenha baixa distorção em um sinal de áudio comprimido, é necessário que a quantidade de bits em sua representação seja aumentada (ISO/IEC, 2004) (PAINTER e SPAN, 2000).

No segundo experimento, o objetivo foi identificar a influência individual dos Códigos de Huffman, sobre a qualidade perceptual (FO) e a taxa de bits. Desta forma, todos os *frames* também foram intencionalmente configurados como *long blocks*, e desta vez, os valores dos *scalefactors* foram mantidos constantes, com valor igual a 90. Os valores dos códigos de Huffman foram alterados entre os seus 12 possíveis valores. Na Tabela 11 são apresentados os valores de distorção do segundo experimento, em sua forma absoluta e em dB, para os dez arquivos de áudio teste.

Índice do Arquivo	Distorção ANMR	Valores dos Códigos de Huffman											
		HCB - 0	HCB - 1	HCB - 2	HCB - 3	HCB - 4	HCB - 5	HCB - 6	HCB - 7	HCB - 8	HCB - 9	HCB - 10	HCB - 11
1	Erro Quant.	175.502	125.006	125.006	87.039	87.039	45.388	45.387	20.147	20.147	13.159	13.159	14.595
	Relação dB	0,00	-2,95	-2,95	-6,09	-6,09	-11,75	-11,75	-18,80	-18,80	-22,50	-22,50	-21,60
2	Erro Quant.	831.499	488.168	488.168	272.882	272.882	109.835	109.838	59.058	58.058	54.853	54.853	55.649
	Relação dB	0,00	-4,63	-4,63	-9,68	-9,68	-17,58	-17,58	-22,97	-23,12	-23,61	-23,61	-23,49
3	Erro Quant.	531.661	309.998	309.998	143.835	143.835	31.455	31.455	23.607	23.607	23.607	23.608	23.608
	Relação dB	0,00	-4,69	-4,69	-11,36	-11,36	-24,56	-24,56	-27,05	-27,05	-27,05	-27,05	-27,05
4	Erro Quant.	410.862	199.693	199.693	108.550	108.550	57.471	57.471	49.249	49.249	49.249	49.250	48.753
	Relação dB	0,00	-6,27	-6,27	-11,56	-11,56	-17,08	-17,08	-18,43	-18,43	-18,43	-18,43	-18,51
5	Erro Quant.	113.472	50.253	50.253	39.298	39.298	36.499	36.499	36.499	36.499	36.514	36.514	36.514
	Relação dB	0,00	-7,07	-7,07	-9,21	-9,21	-9,85	-9,85	-9,85	-9,85	-9,85	-9,85	-9,85
6	Erro Quant.	155.029	66.909	66.909	44.429	44.429	38.401	38.407	38.270	38.270	38.755	38.755	38.755
	Relação dB	0,00	-7,30	-7,30	-10,85	-10,85	-12,12	-12,12	-12,15	-12,15	-12,04	-12,04	-12,04
7	Erro Quant.	18.284	9.000	9.000	8.580	8.580	8.578	8.578	8.578	8.580	8.599	8.599	8.599
	Relação dB	0,00	-6,16	-6,16	-6,57	-6,57	-6,57	-6,57	-6,57	-6,57	-6,55	-6,55	-6,55
8	Erro Quant.	230.612	95.311	95.311	49.428	49.428	34.264	34.264	33.597	33.597	33.599	33.599	33.599
	Relação dB	0,00	-7,67	-7,67	-13,38	-13,38	-16,56	-16,56	-16,73	-16,73	-16,73	-16,73	-16,73
9	Erro Quant.	1.184.797	608.589	608.589	320.937	320.937	157.289	157.289	145.472	145.472	135.121	135.121	135.586
	Relação dB	0,00	-5,79	-5,79	-11,34	-11,34	-17,54	-17,54	-18,22	-18,22	-18,86	-18,86	-18,83
10	Erro Quant.	315.204	139.848	139.848	57.212	57.212	32.635	32.631	32.079	32.080	32.092	32.092	32.242
	Relação dB	0,00	-7,06	-7,06	-14,82	-14,82	-19,70	-19,70	-19,85	-19,85	-19,84	-19,84	-19,80

Tabela 11 – Análise dos Códigos de Huffman vs ANMR

Através da Tabela 11 pode-se verificar que os valores de distorção se mantêm constantes para os *HCBS* das faixas destacadas na tabela, *HCBS-0*; *HCBS-1* e *HCBS-2*; *HCBS-3* e *HCBS-4*; *HCBS-5* e *HCBS-6*; *HCBS-7* e *HCBS-8*; *HCBS-9* e *HCBS-10*; e *HCBS-11*. Este comportamento era esperado, devido aos valores de *LAV* (*Largest Absolute Value*), apresentados na seção 3.2.7, serem os mesmos dentro das faixas de *HCBS* especificadas (ISO/IEC, 2004).

Na Tabela 12, como parte do segundo experimento, são apresentados os valores da taxa de bits obtidos, com os doze diferentes valores de *HCBS*. Desta forma, foi possível realizar uma avaliação da influência individual dos *HCBS* sobre a taxa de bits do sinal de áudio codificado.

Taxa de bits (bps)												
Índice do Arquivo	Valores dos Códigos de Huffman											
	HCBS - 0	HCBS - 1	HCBS - 2	HCBS - 3	HCBS - 4	HCBS - 5	HCBS - 6	HCBS - 7	HCBS - 8	HCBS - 9	HCBS - 10	HCBS - 11
1	1.211	7.655	19.676	7.736	25.806	14.202	48.971	14.047	63.237	11.599	134.492	90.552
2	1.646	12.079	24.486	12.950	31.909	19.538	59.239	19.691	72.812	19.697	134.735	92.382
3	1.394	8.392	18.773	8.937	24.663	14.427	46.935	14.544	58.766	14.659	134.819	91.408
4	1.792	13.367	27.988	13.948	36.256	21.398	67.380	21.547	82.694	21.401	135.079	92.547
5	1.722	11.656	26.829	11.818	34.832	19.448	66.295	19.591	82.673	19.091	135.466	91.974
6	1.655	10.991	24.091	11.221	31.213	17.817	58.753	17.891	73.000	17.934	134.554	91.534
7	1.136	5.245	12.621	5.265	16.381	8.472	29.667	8.504	37.906	8.844	134.482	90.518
8	1.681	11.531	25.069	11.927	32.541	18.743	60.947	18.856	75.608	18.862	134.518	91.648
9	1.651	17.753	27.892	19.279	35.507	24.778	60.697	25.136	74.263	25.083	130.270	92.299
10	1.738	12.635	28.773	13.195	37.738	21.351	57.266	17.729	71.229	17.820	134.824	91.760

Tabela 12 – Análise dos Códigos de Huffman vs taxa de bits

Analisando a Tabela 12 pode-se verificar a tendência de crescimento da taxa de bits, com o aumento dos valores dos *HCBS*. Porém, deve-se observar que para cada uma das faixas de *HCBS* descritas acima, com exceção dos *HCBS-0* e *HCBS-11*, existem dois valores distintos de taxa de bits, um menor e outro maior. Na realidade são duas tentativas de configuração do Codificador de Entropia através dos Códigos de Huffman, para cada valor de *LAV*, com exceção dos *LAVs* de *HCBS-0* e *HCBS-11* que possuem apenas uma configuração. Este comportamento era esperado conforme descrito na norma do codificador MPEG-4 AAC (ISO/IEC, 2004). Para identificar o

comportamento da qualidade perceptual e da taxa de bits, em função dos valores dos Códigos de Huffman, são apresentadas na

Figura 50 duas curvas. Os gráficos se referem ao ensaio realizado com o arquivo de áudio teste de número 9, apresentado nas Tabela 11 e Tabela 12.

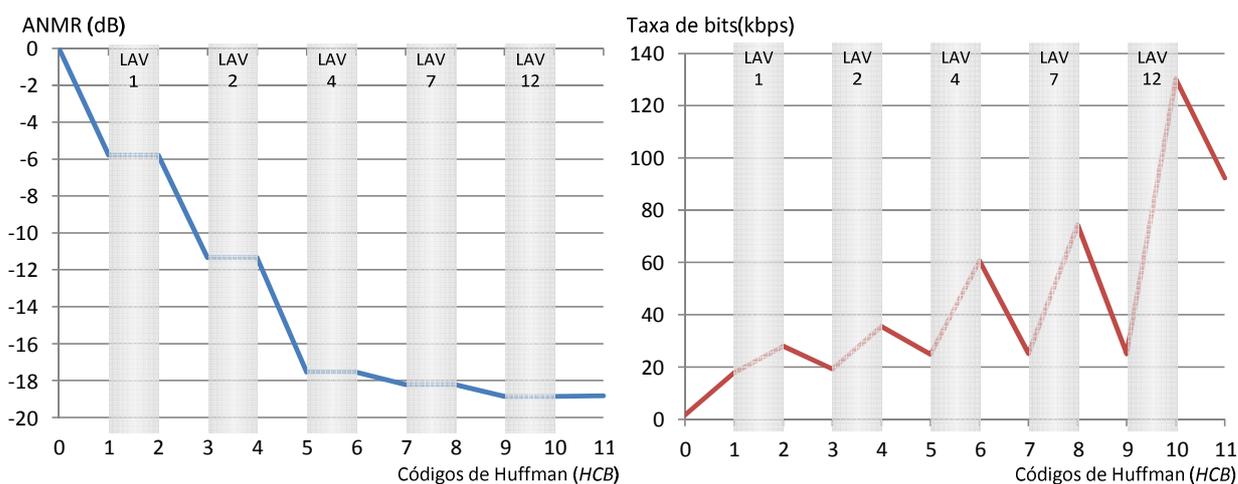


Figura 50 – Comportamento dos Códigos de Huffman vs ANMR e taxa de bits

Através da análise gráfica, as faixas de LAVs dos códigos de Huffman são claramente identificadas. Devido o LAV ser o mesmo para determinados Códigos de Huffman, a qualidade perceptual não é alterada nestes casos (ISO/IEC, 2004). Porém, conforme o LAV vai aumentando, a qualidade perceptual também aumenta, pois com maiores valores para representar os coeficientes espectrais, melhor é sua quantização e menor a distorção do sinal de áudio. O maior impacto dos Códigos de Huffman é sobre a taxa de bits, que é alterada para cada um dos HCBs (ISO/IEC, 2004). Conforme o LAV aumenta, a taxa de bits também tende a aumentar, pois com a possibilidade de representar coeficientes de maior valor, uma quantidade maior de bits acaba sendo necessária. Através do gráfico da taxa de bits, ainda são possíveis de verificar as duas configurações para as curvas de distribuição de probabilidade, para cada um dos LAV (salvo as duas exceções HCB-0 e HCB-11), que podem ser escolhidas para diminuir a quantidade de bits no processo de codificação dos coeficientes espectrais quantizados.

O terceiro experimento foi executado com o objetivo de identificar a influência do agrupamento dos blocos em seções, de um *frame* configurado como *short block*,

sobre a métrica *ANMR* e a taxa de bits. Desta forma, todos os *frames* foram intencionalmente configurados como *short blocks*, e os valores dos *scalefactors* e códigos de Huffman foram mantidos fixados em 90 e 5, respectivamente. Neste experimento, foram utilizadas sete combinações de seções, sendo que o número de seções foi sendo incrementado. Na Figura 51 são apresentados os sete formatos de agrupamento dos blocos utilizados no terceiro experimento de validação, conforme a codificação descrita na seção 6.3.1.1.

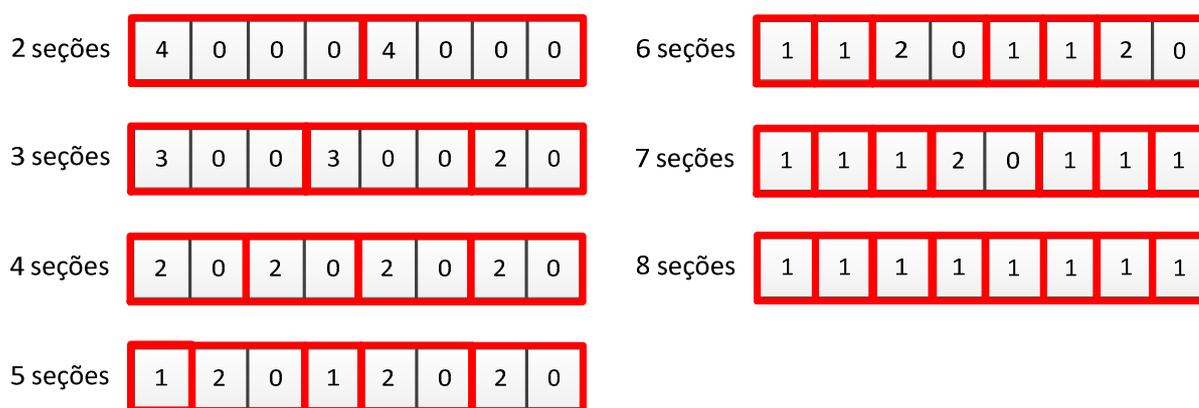


Figura 51 – Configurações de seções para o experimento de validação

Desta forma, os resultados obtidos para distorção, em sua forma absoluta e em dB, utilizando as sete seções ilustradas na Figura 51, para os dez arquivos de áudio teste, podem ser visualizados na Tabela 13. Através de sua análise, identifica-se para os dez arquivos de áudio teste utilizados, que com um aumento na quantidade de seções dentro de um *frame*, há um aumento também na qualidade perceptual do sinal de áudio, ou seja, uma diminuição da distorção. Este comportamento era esperado conforme a normatização do codificador MPEG-4 AAC (ISO/IEC, 2004) (ISO/IEC, 2005), pois com uma quantidade maior de *SFBs*, melhor o sinal de áudio pode ser representado no domínio da frequência, e assim menor a sua distorção.

Índice do Arquivo	Distorção ANMR	Número de Seções						
		2	3	4	5	6	7	8
1	Erro Quant.	16.337	15.349	11.955	9.193	8.262	4.835	3.265
	Relação dB	0,00	-0,54	-2,71	-4,99	-5,92	-10,58	-13,99
2	Erro Quant.	46.607	38.634	32.919	25.880	20.778	10.411	7.460
	Relação dB	0,00	-1,63	-3,02	-5,11	-7,02	-13,02	-15,91
3	Erro Quant.	24.571	17.059	12.033	7.800	5.318	1.134	330
	Relação dB	0,00	-3,17	-6,20	-9,97	-13,29	-26,72	-37,44
4	Erro Quant.	27.003	22.717	18.991	14.703	11.530	4.408	2.049
	Relação dB	0,00	-1,50	-3,06	-5,28	-7,39	-15,74	-22,40
5	Erro Quant.	28.462	23.020	19.563	17.938	16.724	13.074	11.290
	Relação dB	0,00	-1,84	-3,26	-4,01	-4,62	-6,76	-8,03
6	Erro Quant.	21.124	18.764	16.565	12.468	12.024	6.024	5.406
	Relação dB	0,00	-1,03	-2,11	-4,58	-4,89	-10,90	-11,84
7	Erro Quant.	2.012	2.010	1.543	1.156	982	588	244
	Relação dB	0,00	-0,01	-2,31	-4,81	-6,23	-10,69	-18,32
8	Erro Quant.	13.218	11.410	10.241	8.332	6.853	2.630	1.741
	Relação dB	0,00	-1,28	-2,22	-4,01	-5,71	-14,02	-17,61
9	Erro Quant.	147.649	132.063	116.238	86.419	77.681	36.173	30.544
	Relação dB	0,00	-0,97	-2,08	-4,65	-5,58	-12,22	-13,69
10	Erro Quant.	36.566	33.481	30.390	25.614	23.352	12.827	11.716
	Relação dB	0,00	-0,77	-1,61	-3,09	-3,90	-9,10	-9,89

Tabela 13 – Análise do agrupamento dos blocos vs ANMR

Com objetivo de identificar a influência do número de seções dentro de um *frame* sobre a taxa de bits, é apresentada a Tabela 14. Nesta é possível identificar, que com um aumento do número de seções dentro de um *frame*, a quantidade de bits também aumenta. Isso ocorre, pois com um número maior de *SFBs*, mais detalhes do sinal de áudio podem ser representados, o que ocasiona em uma maior quantidade de bits para representar os coeficientes espectrais, e também uma quantidade de bits maior para representar os dados de informações auxiliares, que são inseridos dentro do fluxo de dados do codificador. Este comportamento era esperado, segundo a documentação da norma do codificador MPEG-4 AAC (ISO/IEC, 2004) (ISO/IEC, 2005).

Taxa de bits (bps)							
Índice do Arquivo	Número de Seções						
	2	3	4	5	6	7	8
1	25.820	34.552	43.429	45.710	45.756	51.728	54.359
2	28.870	36.585	44.157	45.852	45.992	49.884	51.972
3	29.609	37.174	44.567	45.622	46.085	49.679	50.578
4	28.821	35.229	41.640	45.892	46.484	49.943	50.615
5	29.154	36.102	41.855	43.573	45.491	50.244	51.037
6	25.702	33.569	40.996	45.095	45.204	50.849	51.783
7	26.214	34.961	43.614	45.409	45.739	50.881	52.495
8	28.831	36.041	43.074	45.902	46.324	49.464	50.930
9	26.021	33.764	41.280	43.800	45.774	48.888	50.224
10	27.756	34.658	41.503	45.703	46.229	49.896	51.378

Tabela 14 – Análise do agrupamento dos blocos vs taxa de bits

Para facilitar a análise, são apresentadas na Figura 52 as curvas que relacionam o comportamento do número de seções dos *frames* com a qualidade perceptual e a taxa de bits. Na elaboração das curvas foi utilizado o resultado obtido do arquivo de áudio teste de número 9, apresentado nas Tabela 13 e Tabela 14.

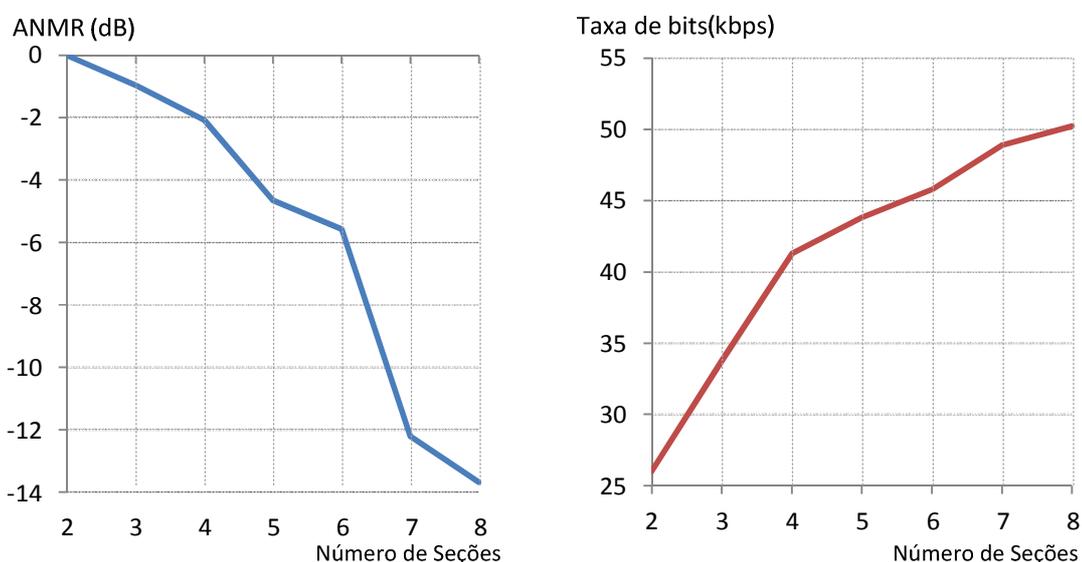


Figura 52 – Comportamento do agrupamento dos blocos vs ANMR e taxa de bits

Através da análise gráfica, observa-se que com o aumento do número de seções dentro de um frame, a distorção tende a ter seu valor diminuído, e conseqüentemente, o valor da taxa de bits é aumentado. É interessante comentar

que o comportamento identificado na Figura 52, também é verificado nos demais arquivos de áudio teste. Outro ponto que deve-se ressaltar, é que esta variável de decisão, apenas influencia no modelo computacional, caso o *frame* seja configurado como *short block*.

Para os três experimentos de validação realizados, analisando individualmente cada uma das variáveis de decisão, pôde-se verificar a relação inversa, entre a distorção do sinal de áudio e sua taxa de bits. Quanto maior a taxa de bits menor a distorção (ISO/IEC, 2004) (BOSI e GOLDBERG, 2003). Verificou-se também a influência das três variáveis de decisão sobre as métricas mais significativas do modelo computacional, a qualidade perceptual e a taxa de bits. Observou-se que a variável *scalefactor* tem maior relevância sobre a distorção do sinal de áudio, pois define os intervalos de quantização dos coeficientes espectrais. As outras duas variáveis, *HCBs* e agrupamento dos blocos, também impactam sobre a distorção, porém sua influência principal é sobre a taxa de bits, através de estratégias de compressão de dados (ISO/IEC, 2004) (ISO/IEC, 2005).

Assim, conforme relatado, identificou-se que a dinâmica das variáveis apresentada pelo modelo computacional desenvolvido para as três variáveis, representa de maneira fidedigna o comportamento do codificador MPEG-4 AAC. Desta forma, baseada na documentação de normatização do codificador MPEG-4 AAC (ISO/IEC, 2004) (ISO/IEC, 2005), o modelo computacional desenvolvido pôde ser validado.

7.3 Experimentos

Nesta seção são apresentados os experimentos realizados com o modelo computacional desenvolvido para otimizar a qualidade perceptual do codificador MPEG-4 AAC. Inicialmente, os experimentos de parametrização das metaheurísticas são abordados, e na sequência os experimentos de aplicação, a fim de confrontar os resultados das metaheurísticas com os obtidos através do codificador de referência. Os experimentos foram realizados em um processador Intel Core i3, sobre o sistema operacional Linux Ubuntu.

7.3.1 Parametrização das Metaheurísticas

Inicialmente, as metaheurísticas foram trabalhadas individualmente, fora da estrutura híbrida, para que se pudesse obter a melhor condição de operação de cada uma. Assim, foi possível analisar o processo de exploração do espaço de soluções de ambas metaheurísticas operando sozinhas e em conjunto. Desta forma, nesta seção, primeiro são apresentados os testes com os parâmetros de calibração do Algoritmo Genético, na sequência da Busca Tabu, e ao final, o processo de calibração da estrutura híbrida com as duas metaheurísticas juntas.

Para analisar o modelo computacional foram determinadas quatro taxas de bits: 48kbps, 116kbps, 184kbps e 250kbps. A menor taxa de bits foi escolhida como 48kbps, pois para taxas abaixo desta, a qualidade perceptual acaba sendo muito baixa. Para a frequência de amostragem de 44.1 kHz dos arquivos de áudio teste, a máxima taxa de bits é 250kbps (ISO/IEC, 2004). Nos experimentos de calibração foi utilizado um único arquivo de áudio teste, o de número 9 apresentado na Tabela 8. Este arquivo foi escolhido devido ao fato de ter apresentado o maior valor de distorção, nas quatro taxas de bits citadas, para a técnica *TLS*, representando assim, uma situação de dificuldade maior para o processo de codificação. Outro motivo da sua escolha foi devido ao seu estilo musical classificado como rock, que apresenta uma grande entropia musical, devido possuir diversos elementos ocorrendo juntos no tempo (TANKANOW, 2007) (SINGH, 2011).

7.3.1.1 Algoritmo Genético

A intenção dos experimentos com o Algoritmo Genético foi avaliar a dinâmica da busca de boas soluções pelo algoritmo, para o Problema de Codificação AAC abordado neste trabalho.

Para que os resultados obtidos através das metaheurísticas, durante o processo de calibração, pudessem ser avaliados, o arquivo de áudio teste de número 9 foi codificado nas quatro taxas de bits citadas, utilizando a técnica *TLS* do codificador de referência. Durante estas codificações foram armazenados os valores de *ANMR* (FO) e tempo de processamento, obtidos através do método *TLS*. Desta forma, estes valores puderam ser utilizados como referência, comparando-os com os

resultados obtidos com as metaheurísticas. Os resultados são apresentados na Tabela 15.

Taxa de bits	48kbps	116kbps	184kbps	250kbps
ANMR (FO) - Erro Quantização	55.476	11.501	1.084	354
Tempo de Processamento (μs)	32.410	34.978	36.910	37.299

Tabela 15 – Codificação do arquivo de áudio 9 utilizando *Two Loop Search*

Nos experimentos de parametrização do Algoritmo Genético foram alterados os valores da Taxa de Mutação e do Número de Gerações, a fim de estudar a influência destes, sobre a Função Objetivo, e o tempo de processamento necessário para a obtenção da solução de um *frame* de áudio. A taxa de cruzamento foi fixada em 90%, o processo de seleção por torneio foi configurado com probabilidade de 80% para os mais aptos, e o tamanho da população foi de 20 indivíduos. Os valores da Taxa de Mutação e Número de Gerações foram alterados conforme a Tabela 16 e Tabela 17, respectivamente.

Taxa de Mutação (%)
10
30
50
70
90

Tabela 16 – Relação dos percentuais para a taxa de mutação

Número de Gerações
10
50
100
200
500

Tabela 17 – Relação dos valores para o Número de Gerações

Desta forma, para contemplar todas as combinações entre os parâmetros, foram realizados 25 experimentos para cada uma das quatro taxas de bits pré-definidas. Devido ao comportamento estocástico das metaheurísticas, cada configuração dos parâmetros foi repetida 50 vezes, de maneira que o valor de FO, e o tempo de processamento (μs) para cada *frame*, foram representados pela média μ e pelo seu desvio padrão σ . Na Tabela 18, são apresentados os resultados obtidos do primeiro experimento com o Algoritmo Genético para o arquivo de áudio teste de número 9, para a taxa de 48kbps. O tempo de processamento é expresso em μs , e para facilitar a identificação, os melhores resultados foram destacados na tabela.

Percentual de Mutação	Parâmetros de Avaliação		Número de Gerações				
			10	50	100	200	500
10%	ANMR - FO	μ	56.533	56.065	55.998	55.796	54.988
		σ	415	387	377	360	312
	Tempo de Processamento (μs)	μ	17.737	83.041	174.117	346.005	824.215
		σ	160	315	551	871	1.007
30%	ANMR - FO	μ	54.925	52.566	52.804	52.506	52.448
		σ	421	392	370	357	359
	Tempo de Processamento (μs)	μ	16.979	84.907	159.956	315.097	748.324
		σ	155	345	526	687	986
50%	ANMR - FO	μ	54.554	55.008	53.175	53.200	52.877
		σ	312	398	474	341	345
	Tempo de Processamento (μs)	μ	16.550	82.719	159.723	320.449	772.897
		σ	148	304	509	622	967
70%	ANMR - FO	μ	53.506	53.833	53.735	54.806	53.188
		σ	309	412	387	351	343
	Tempo de Processamento (μs)	μ	16.487	81.099	161.980	355.091	794.745
		σ	139	374	549	780	986
90%	ANMR - FO	μ	54.043	53.801	53.717	53.884	54.541
		σ	379	409	398	377	328
	Tempo de Processamento (μs)	μ	16.757	79.067	163.007	371.095	806.392
		σ	157	358	648	817	1.098

Tabela 18 – Resultados do primeiro experimento do AG para a taxa de bits de 48kbps

Através da Tabela 18, pode-se verificar que a melhor solução do Algoritmo Genético, obteve uma melhora de aproximadamente 5%, em relação ao valor de ANMR da técnica TLS. Isso significa uma melhora de 5% na qualidade perceptual do sinal de áudio codificado. Porém, deve ser observado que o tempo processamento

foi consideravelmente superior ao necessário pela técnica *TLS*. Também deve ser destacado, que os melhores resultados foram obtidos com percentual de mutação igual a 30%. Isso demonstra, neste caso, que um excesso da estratégia de diversificação na busca conduziu as soluções para espaços de soluções não tão promissores.

Na Tabela 19, são apresentados os resultados obtidos através do Algoritmo Genético, para a taxa de 116kbps. Nestes experimentos, o arquivo de áudio teste de número 9 também foi codificado por 50 vezes, para cada uma das combinações dos parâmetros, de maneira que os resultados foram representados por uma média μ e pelo seu desvio padrão σ .

Percentual de Mutação	Parâmetros de Avaliação		Número de Gerações				
			10	50	100	200	500
10%	ANMR - FO	μ	52.959	45.771	40.214	33.905	29.409
		σ	1.298	703	698	495	463
	Tempo de Processamento (μ s)	μ	20.905	89.165	203.624	496.172	974.952
		σ	170	378	488	517	2.576
30%	ANMR - FO	μ	36.866	21.981	21.432	21.378	20.082
		σ	1.305	806	583	426	398
	Tempo de Processamento (μ s)	μ	20.010	85.227	190.280	37.410	933.605
		σ	176	372	476	537	2.365
50%	ANMR - FO	μ	29.834	25.179	22.844	22.537	22.195
		σ	1.286	798	559	509	496
	Tempo de Processamento (μ s)	μ	19.554	86.130	189.065	378.146	932.623
		σ	165	389	427	796	2.443
70%	ANMR - FO	μ	32.202	27.961	24.061	24.912	23.779
		σ	1.396	606	569	426	376
	Tempo de Processamento (μ s)	μ	19.869	91.067	189.698	375.168	944.958
		σ	159	382	429	765	2.089
90%	ANMR - FO	μ	33.143	30.486	26.404	26.678	26.215
		σ	1.276	804	593	573	485
	Tempo de Processamento (μ s)	μ	19.671	90.663	191.806	387.064	950.193
		σ	162	386	396	716	2.358

Tabela 19 – Resultados do primeiro experimento do AG para a taxa de bits de 116kbps

Analisando a Tabela 19 e comparado com a Tabela 15, é possível identificar que a melhor solução encontrada pelo Algoritmo Genético possui valor de *ANMR* aproximadamente 74% superior à encontrada pela técnica *TLS*. Se tratando de um

problema de minimização, este resultado é considerado muito ruim. De maneira semelhante ao comportamento para a taxa de 48kbps, os melhores resultados foram obtidos com uma taxa de mutação de 30%, para diferentes valores de Número de Gerações.

Para a taxa de 184kbps, os mesmos experimentos foram realizados. O arquivo de áudio teste de número 9 foi codificado 50 vezes, para cada uma das 25 combinações dos parâmetros. Desta forma, na Tabela 20 são apresentados os valores de qualidade perceptual (*ANMR*) e tempo de processamento. Os valores são representados por uma média μ e desvio padrão σ .

Percentual de Mutação	Parâmetros de Avaliação		Número de Gerações				
			10	50	100	200	500
10%	ANMR - FO	μ	45.446	21.428	16.310	14.632	11.701
		σ	1.796	1.546	1.466	1.202	988
	Tempo de Processamento (μ s)	μ	21.890	106.215	207.184	398.225	989.061
		σ	259	344	428	578	879
30%	ANMR - FO	μ	16.761	6.698	6.513	6.388	6.080
		σ	607	560	512	459	288
	Tempo de Processamento (μ s)	μ	19.781	95.858	187.979	401.021	944.850
		σ	370	387	405	584	1.021
50%	ANMR - FO	μ	10.871	8.628	8.524	8.304	8.137
		σ	343	462	510	474	325
	Tempo de Processamento (μ s)	μ	19.488	95.884	188.065	385.660	945.660
		σ	285	361	493	730	1.844
70%	ANMR - FO	μ	12.806	9.717	9.875	9.503	8.826
		σ	814	765	886	536	378
	Tempo de Processamento (μ s)	μ	19.282	95.859	189.208	403.496	945.300
		σ	395	453	546	808	2.263
90%	ANMR - FO	μ	13.995	13.187	12.020	11.813	9.998
		σ	1.187	1.020	1.165	844	412
	Tempo de Processamento (μ s)	μ	19.162	98.029	189.728	407.885	950.390
		σ	386	369	470	1.161	5.669

Tabela 20 – Resultados do primeiro experimento do AG para a taxa de bits de 184kbps

Comparando-se o melhor valor obtido com o Algoritmo Genético e o valor obtido com a técnica *TLS*, pode-se verificar que o valor de *ANMR* obtido pelo AG é 460% superior ao valor de *ANMR* encontrado pelo método do codificador de referência. Isso representa uma diminuição de 460% na qualidade perceptual do

sinal de áudio codificado com o Algoritmo Genético, em relação à codificação realizada pelo método *TLS*.

O último ensaio realizado, para o primeiro experimento com o Algoritmo Genético, foi utilizando a taxa de 250kbps. Foram realizados 50 codificações, para cada uma das 25 combinações dos parâmetros das metaheurísticas, do arquivo de áudio teste de número 9. Os resultados foram representados por uma média μ e desvio padrão σ , e são apresentados na Tabela 21.

Percentual de Mutação	Parâmetros de Avaliação		Número de Gerações				
			10	50	100	200	500
10%	ANMR - FO	μ	30.802	19.556	6.025	5.578	3.742
		σ	1.201	1.103	680	598	324
	Tempo de Processamento (μ s)	μ	21.900	185.662	219.833	477.019	1.048.813
		σ	220	2.487	3.498	6.103	37.150
30%	ANMR - FO	μ	9.852	5.054	2.414	2.460	2.309
		σ	1.022	973	589	443	301
	Tempo de Processamento (μ s)	μ	19.992	175.469	196.325	349.775	986.650
		σ	198	2.397	2.766	5.830	35.981
50%	ANMR - FO	μ	4.960	4.101	3.576	3.498	3.222
		σ	635	512	483	309	276
	Tempo de Processamento (μ s)	μ	19.440	168.223	193.797	322.197	962.454
		σ	167	2.347	2.673	5.509	27.201
70%	ANMR - FO	μ	4.145	4.033	3.990	3.707	3.503
		σ	487	374	297	220	196
	Tempo de Processamento (μ s)	μ	19.182	180.273	193.542	356.197	961.493
		σ	183	2.574	2.945	6.014	24.990
90%	ANMR - FO	μ	4.004	3.842	3.672	3.702	3.512
		σ	387	412	287	215	166
	Tempo de Processamento (μ s)	μ	19.397	196.301	193.233	374.119	961.080
		σ	173	2.189	3.278	6.449	31.440

Tabela 21 – Resultados do primeiro experimento do AG para a taxa de bits de 250kbps

Em uma análise da Tabela 21, novamente verifica-se que a solução encontrada pelo Algoritmo Genético tem uma qualidade inferior, quando comparada à encontrada pelo método *TLS* do codificador de referência. Neste caso, o valor de *ANMR* encontrado pelo Algoritmo Genético foi aproximadamente 552% superior ao valor encontrado pela técnica do codificador padrão.

Através dos resultados do primeiro experimento com o Algoritmo Genético, apresentados nas Tabelas 18, 19, 20 e 21, em uma análise das quatro taxas de bits,

foi observado que em apenas uma das taxas (48kbps) é que o Algoritmo Genético conseguiu obter uma solução de melhor qualidade do que a solução do algoritmo *TLS* apresentada na Tabela 15. Para as outras três taxas, as soluções foram consideravelmente piores do que as encontradas pelo método do codificador de referência apresentadas também na Tabela 15. Diante desta ineficiência do Algoritmo Genético em produzir baixos valores para a distorção *ANMR*, foi necessário alterar a estrutura inicial do algoritmo.

Através do estudo abordado na seção de Validação (7.2), identificou-se que a variável que impactou com maior intensidade sobre a distorção foi o *scalefactor*. Desta forma, a fim de identificar o motivo do insucesso do AG, foi realizado um estudo de análise do comportamento dos *scalefactors*, em função de cada *scalefactor band*, para o Algoritmo *Two Loop Search* e o Algoritmo Genético. Para realização desta análise, foi elaborado um gráfico que relacionou os valores dos *scalefactors* de cada uma das *scalefactor bands*, para as quatro taxas de bits estudadas. O *frame* de número 300 do arquivo de áudio teste de número 9 foi determinado como padrão para o estudo. Assim, a relação entre *scalefactors* e *scalefactor bands* para o algoritmo *TLS* do codificador de referência é apresentada na Figura 53.

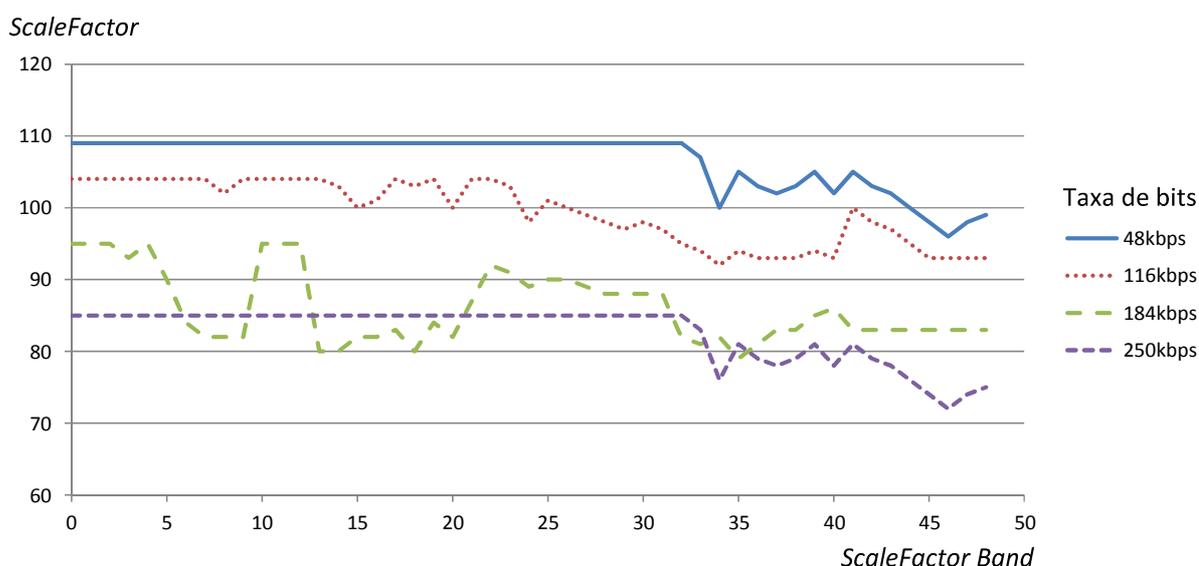


Figura 53 – Dinâmica entre *scalefactors* e *scalefactor bands* obtida com o codificador de referência (*TLS*)

Pode-se observar na Figura 53, que os valores de *scalefactors* das *scalefactors bands*, de uma mesma taxa de bits, não variam muito entre si. Este fato pode ser também verificado na Tabela 22. Nesta são apresentados os valores de média (μ) e desvio padrão (σ) dos *scalefactors* do *frame* de estudo, obtidos através da codificação realizada pela técnica *TLS*, para as quatro taxas de bits.

Valores dos <i>ScaleFactors</i>	Taxa de bits			
	48kbps	116kbps	184kbps	250kbps
μ	107	99	86	83
σ	3,79	4,39	4,95	3,79

Tabela 22 – Média e desvio padrão dos *scalefactors* obtidos com o codificador de referência (*TLS*)

Através da Tabela 22, é possível constatar que o desvio padrão dos valores de *scalefactors* para as quatro taxas, não é elevado, de maneira que existe uma tendência dos valores de *scalefactors* dentro do *frame*.

Com o objetivo de comparar este comportamento dos *scalefactors*, obtidos através da codificação utilizando a técnica *TLS*, com o comportamento do Algoritmo Genético, este mesmo experimento foi realizado. De maneira análoga, os valores de *scalefactors* de cada *scalefactor band* foram armazenados, para que o gráfico da Figura 54 fosse elaborado. Nesta ilustração foi utilizado *frame* de número 300 do arquivo de áudio teste de número 9.

Pode-se verificar na Figura 54 uma dinâmica totalmente diferente para os *scalefactors*, quando comparada ao comportamento destes, obtido através do método *TLS*. Neste experimento, os *scalefactors* variaram muito entre si. É possível identificar para a taxa de 48kbps, que os valores de *scalefactors* variam de seu valor mínimo até o seu máximo valor, excursionado em todo o seu *range*.

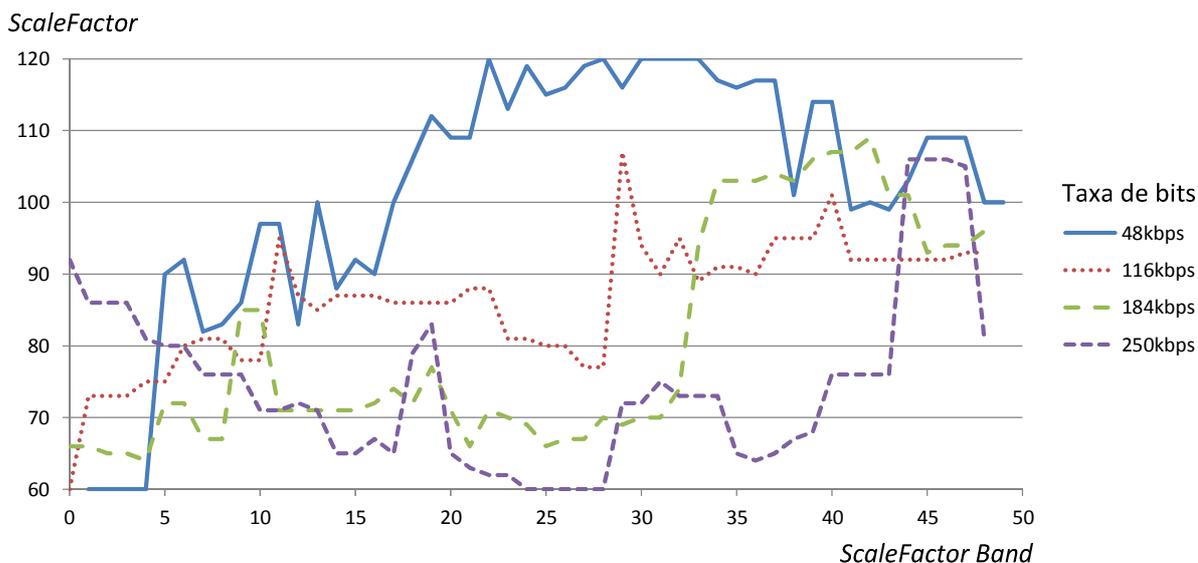


Figura 54 – Dinâmica entre *scalefactors* e *scalefactor bands* obtida com o primeiro experimento do Algoritmo Genético

Ao representar os valores de *scalefactors* através de uma média e desvio padrão, verifica-se exatamente esta alta variação de seus valores, dentro do *frame* codificado utilizando o Algoritmo Genético. Desta forma, são apresentados na Tabela 23 os valores obtidos de média e desvio padrão para as quatro taxas de bits. Devido os valores terem sido gerados pela metaheurística AG, estes experimentos foram repetidos 50 vezes.

Valores dos <i>ScaleFactors</i>	Taxa de bits			
	48kbps	116kbps	184kbps	250kbps
μ	102	86	80	75
σ	16,86	8,42	15,20	12,15

Tabela 23 – Média e desvio padrão dos *scalefactors* obtidos com o primeiro experimento do Algoritmo Genético

Observa-se na Tabela 23, que os valores de desvio padrão dos *scalefactors*, obtidos pela codificação utilizando o Algoritmo Genético, para as quatro taxas de bits estudadas, são superiores aos valores de desvio padrão dos *scalefactors* obtidos com a técnica *TLS* do codificador de referência. Isso demonstra que os valores de

scalefactors, neste caso do Algoritmo Genético, foram mais espaçados, com uma tendência de valores menor do que à obtida com a técnica *TLS* do codificador de referência.

Através de uma análise em conjunto, das Figura 53 e Figura 54 e das Tabela 22 e Tabela 23, pode-se identificar que os valores dos *scalefactors* definidos pela técnica *TLS* não variam muito entre si. A média do desvio padrão dos *scalefactors* neste caso é de apenas 4,23. Porém, os valores de *scalefactors* encontrados pelo Algoritmo Genético variam mais entre si, com média do desvio padrão igual a 13,15. Este excesso de variação entre os *scalefactors* de um mesmo *frame* é provocado pelo operador de mutação do algoritmo, que insere muita diversidade de valores dentro do *frame*. Como os *scalefactors* são codificados diferencialmente, um excesso de variação entre os mesmos, aumenta a quantidade de bits necessários em sua representação. Isso provoca um aumento da taxa de bits total, e assim as soluções viáveis são mais difíceis de serem geradas, sendo mais facilmente restringidas pelo modelo computacional, devido atingirem com maior facilidade a máxima taxa de bits permitida.

Os valores dos *scalefactors* são alterados pelo Método de Mutação 1, representado na Figura 44, que altera aleatoriamente seus valores atendendo às restrições específicas da variável de decisão. Deve-se observar que devido os *SFs* serem codificados diferencialmente, a diferença entre dois *SFs* adjacentes jamais deve ser superior ao valor 60 (ISO/IEC, 2004).

A fim de evitar que a diferença entre os valores dos *scalefactors* dentro de um *frame* fosse elevada, a técnica de alteração dos *scalefactors* do Método de Mutação 1 foi modificada. Assim, ela deixou de ser totalmente aleatória, podendo assumir qualquer valor válido para um *scalefactor*, e passou a ser restringida em 3 unidades para mais ou para menos, a partir do valor atual do *scalefactor*. Esta técnica de mutação foi testada, utilizando o arquivo de áudio teste de número 9. Desta forma, o comportamento identificado entre *scalefactors* e *scalefactor bands* é apresentado na Figura 55.

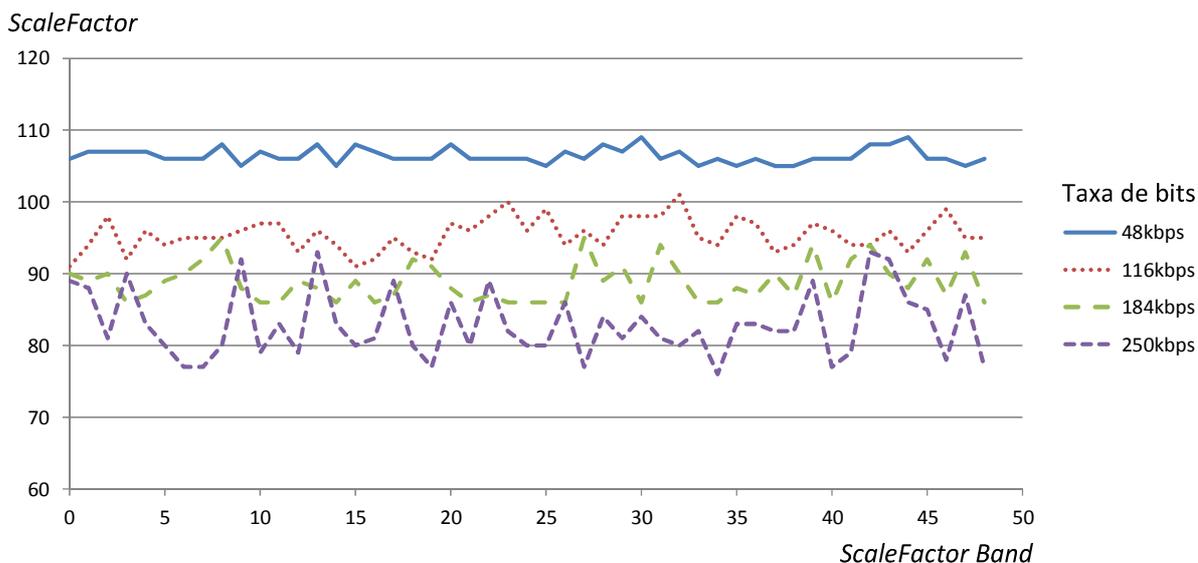


Figura 55 – Dinâmica entre *scalefactors* e *scalefactor bands* obtida com o segundo experimento do Algoritmo Genético

Através da Figura 55, é possível identificar que com a estratégia de restrição dos valores de *scalefactors*, inserida no Método de Mutação 1, seus valores variaram menos entre si, para uma determinada taxa de bits. Este comportamento demonstra uma tendência entre os valores dos *scalefactors*. Esta tendência, também pode ser verificada através da Tabela 24, que apresenta os valores de média e desvio padrão dos *scalefactors*, obtidos com o segundo experimento do Algoritmo Genético para as quatro taxas de bits.

Valores dos <i>ScaleFactors</i>	Taxa de bits			
	48kbps	116kbps	184kbps	250kbps
μ	106	95	89	83
σ	1,05	2,26	2,77	4,65

Tabela 24 – Média e desvio padrão dos *scalefactors* obtidos com o segundo experimento do Algoritmo Genético

Com a alteração do Método de Mutação 1 para os *scalefactors*, seus valores ficaram mais próximos, com a média de desvio padrão igual a 2,68. Era justamente este comportamento que se necessitava sobre os *scalefactors*, para que não ultrapassassem com facilidade a máxima taxa de bits, e as soluções não fossem

inviáveis. Desta forma, para avaliar de que maneira esta alteração impactou sobre a qualidade das soluções, os experimentos para avaliação de FO e tempo de processamento foram refeitos para as quatro taxas de bits. Na Tabela 25 são apresentados os resultados obtidos para a taxa de 48kbps. Cada uma das 25 combinações entre os parâmetros do Algoritmo Genético foi realizada 50 vezes, e desta forma, foram representados por uma média μ e desvio padrão σ .

Percentual de Mutação	Parâmetros de Avaliação		Número de Gerações				
			10	50	100	200	500
10%	ANMR - FO	μ	40.622	35.581	35.622	35.617	35.617
		σ	986	405	354	225	182
	Tempo de Processamento (μ s)	μ	8.034	49.765	111.107	274.426	565.845
		σ	381	925	2.016	3.604	3.691
30%	ANMR - FO	μ	38.445	31.859	31.928	31.903	34.147
		σ	789	214	181	142	140
	Tempo de Processamento (μ s)	μ	8.599	54.466	113.976	231.494	601.447
		σ	379	946	2.141	3.067	13.712
50%	ANMR - FO	μ	32.346	31.789	31.764	31.753	31.907
		σ	543	220	148	136	101
	Tempo de Processamento (μ s)	μ	9.150	56.106	115.731	319.071	655.438
		σ	378	1.013	2.422	4.506	28.879
70%	ANMR - FO	μ	31.978	31.520	31.547	31.513	31.017
		σ	512	205	185	155	150
	Tempo de Processamento (μ s)	μ	9.311	56.953	115.520	243.259	721.850
		σ	315	882	2.792	5.728	61.903
90%	ANMR - FO	μ	30.661	30.405	30.522	30.524	30.775
		σ	478	126	182	151	119
	Tempo de Processamento (μ s)	μ	9.728	57.491	117.746	252.621	791.773
		σ	291	860	3.082	5.866	84.415

Tabela 25 – Resultados do segundo experimento do AG para a taxa de bits de 48kbps

Através da análise da Tabela 25, pode-se identificar que os resultados obtidos com o segundo experimento do AG, foram muito melhores do que os resultados do primeiro. Além disso, foram melhores do que os resultados obtidos pela técnica *TLS* do codificador de referência. O valor de *ANMR* encontrado pelo Algoritmo Genético foi aproximadamente 1,82 vezes menor do que o encontrado pela técnica do codificador de referência.

Deve-se observar, que com a alteração no Método de Mutação 1, o Algoritmo Genético passou a ter um comportamento totalmente diferente. Verifica-se que no segundo experimento do AG, as melhores soluções foram obtidas com 90% de taxa de mutação, enquanto que no primeiro experimento este percentual foi de 30%. Como apresentado na Figura 44, o percentual de mutação atua sobre a quantidade de *SFBs* que terão seus valores de *SF* e *HCb* alterados. Desta forma, com a restrição dos valores dos *scalefactors*, o processo de busca identificou que boas soluções foram alcançadas, alterando pouco os valores de *scalefactors*, porém uma quantidade maior de *SFBs*. Por isso o aumento do percentual de mutação.

Na Tabela 26 são apresentados os resultados do segundo experimento com o Algoritmo Genético para a taxa de 116kbps. Os experimentos foram realizados por 50 vezes de maneira que os valores foram representados por uma média μ e desvio padrão σ .

Percentual de Mutação	Parâmetros de Avaliação		Número de Gerações				
			10	50	100	200	500
10%	ANMR - FO	μ	38.195	20.239	18.302	18.027	17.778
		σ	1.843	1.077	611	525	274
	Tempo de Processamento (μ s)	μ	8.269	53.001	128.118	336.303	865.739
		σ	281	1.085	1.974	4.527	64.732
30%	ANMR - FO	μ	27.498	6.053	6.083	6.031	5.974
		σ	1.198	258	258	161	155
	Tempo de Processamento (μ s)	μ	8.674	63.778	148.775	322.984	876.113
		σ	335	1.484	1.972	4.071	52.156
50%	ANMR - FO	μ	16.811	4.709	4.642	4.700	4.686
		σ	983	196	116	101	53
	Tempo de Processamento (μ s)	μ	9.471	69.907	163.240	327.770	927.715
		σ	460	1.217	2.107	3.977	13.694
70%	ANMR - FO	μ	13.653	4.179	4.192	4.207	3.986
		σ	884	89	92	59	55
	Tempo de Processamento (μ s)	μ	10.079	73.822	157.759	329.241	925.441
		σ	1.348	3.017	2.827	4.865	12.930
90%	ANMR - FO	μ	7.995	3.751	3.749	3.753	3.765
		σ	733	74	55	42	33
	Tempo de Processamento (μ s)	μ	14.626	75.915	159.227	338.905	937.389
		σ	6.552	3.011	3.493	5.253	11.102

Tabela 26 – Resultados do segundo experimento do AG para a taxa de bits de 116kbps

Analisando a Tabela 26 pode-se identificar que o melhor resultado de *ANMR* obtido pelo Algoritmo Genético é aproximadamente 3 vezes inferior ao resultado obtido com a técnica do codificador de referência. Isso significa uma considerável melhora na qualidade perceptual do sinal de áudio codificado. Verifica-se também, conforme observado para a taxa de 48kbps, a influência da alteração do Método de Mutação 1, de maneira que os melhores resultados foram encontrados com percentual de mutação igual a 90%. Em relação ao tempo de processamento, identifica-se um aumento de aproximadamente 120ms, no tempo necessário para obter a melhor solução do Algoritmo Genético em relação à solução encontrada pela técnica *TLS*.

Para a taxa de 184kbps, os experimentos para as 25 combinações dos parâmetros do Algoritmo Genético, foram repetidos. Desta forma, na Tabela 27 são apresentados os resultados para 50 execuções de cada combinação, de maneira que os valores de FO e tempo de processamento são representados por uma média μ e desvio padrão σ .

Percentual de Mutação	Parâmetros de Avaliação		Número de Gerações				
			10	50	100	200	500
10%	ANMR - FO	μ	38.670	19.783	16.433	16.247	15.080
		σ	1.686	1.097	840	488	297
	Tempo de Processamento (μ s)	μ	8.521	52.491	151.596	358.112	992.742
		σ	332	1.232	5.496	4.986	7.243
30%	ANMR - FO	μ	20.991	3.153	2.761	2.754	2.640
		σ	1.599	476	312	211	200
	Tempo de Processamento (μ s)	μ	8.612	64.517	159.680	394.831	1.055.693
		σ	402	8.329	2.318	5.174	9.985
50%	ANMR - FO	μ	17.266	1.250	1.262	1.253	1.252
		σ	1.233	166	144	109	61
	Tempo de Processamento (μ s)	μ	9.409	77.327	186.069	414.339	1.063.852
		σ	418	18.727	2.029	5.323	13.171
70%	ANMR - FO	μ	10.932	906	924	919	920
		σ	955	110	122	67	65
	Tempo de Processamento (μ s)	μ	12.964	82.263	190.469	417.830	1.125.693
		σ	1.277	9.922	3.321	6.289	13.255
90%	ANMR - FO	μ	7.772	693	702	704	705
		σ	531	71	63	44	31
	Tempo de Processamento (μ s)	μ	16.832	83.943	197.660	427.285	1.164.482
		σ	6.553	1.368	4.309	6.038	14.873

Tabela 27 – Resultados do segundo experimento do AG para a taxa de bits de 184kbps

Através da Tabela 27, pode-se verificar que as melhores soluções encontradas pelo Algoritmo Genético neste segundo experimento, são de maior qualidade do que a solução encontrada pela técnica *TLS*. A melhor solução encontrada pelo AG, possui o valor de FO aproximadamente 1,56 vezes menor, que o valor de FO da solução encontrada pelo algoritmo *TLS* do codificador de referência. Novamente, pode-se verificar o impacto no algoritmo da alteração do Método de Mutação 1, de maneira que as melhores soluções foram encontradas com percentual de mutação de 90%. Pode-se identificar também, que o tempo de processamento necessário para obter a melhor solução do Algoritmo Genético, é aproximadamente 45ms superior, ao tempo necessário pela técnica *TLS*.

Na última etapa do segundo experimento, foram realizadas as codificações para as 25 combinações dos parâmetros do Algoritmo Genético, utilizando a taxa de 250kbps. Foram realizadas 50 execuções para cada combinação, de maneira que os valores de *ANMR* e tempo de processamento foram representados por uma média μ e desvio padrão σ . Os resultados são apresentados na Tabela 28.

Percentual de Mutação	Parâmetros de Avaliação		Número de Gerações				
			10	50	100	200	500
10%	ANMR - FO	μ	38.795	19.978	16.704	15.124	13.223
		σ	1.356	1.038	667	411	321
	Tempo de Processamento (μ s)	μ	8.328	52.402	121.329	402.983	1.090.757
		σ	270	1.262	8.055	8.915	17.734
30%	ANMR - FO	μ	22.964	3.102	2.001	1.952	1.887
		σ	1.164	445	271	148	135
	Tempo de Processamento (μ s)	μ	8.947	61.495	175.060	456.607	1.198.556
		σ	268	1.450	2.829	8.103	28.457
50%	ANMR - FO	μ	17.282	732	643	585	572
		σ	1.084	263	162	89	55
	Tempo de Processamento (μ s)	μ	9.301	83.924	207.413	484.889	1.367.118
		σ	272	1.241	2.572	5.018	59.608
70%	ANMR - FO	μ	12.058	381	347	342	337
		σ	956	177	96	66	56
	Tempo de Processamento (μ s)	μ	13.557	85.471	222.828	495.016	1.367.955
		σ	3.450	1.498	3.332	6.261	45.712
90%	ANMR - FO	μ	7.882	223	218	214	200
		σ	592	98	59	42	24
	Tempo de Processamento (μ s)	μ	15.582	87.902	231.385	502.224	1.368.846
		σ	7.252	1.259	3.554	6.478	38.237

Tabela 28 – Resultados do segundo experimento do AG para a taxa de bits de 250kbps

Analisando a Tabela 28, novamente identifica-se, conforme os experimentos das demais taxas de bits, que o Algoritmo Genético obteve melhores resultados quando comparado ao resultado do método *TLS* do codificador de referência. Para a taxa de 250kbps, a redução obtida pelo Algoritmo Genético do valor de FO, foi de aproximadamente 1,77 vezes em relação ao valor obtido com o algoritmo *TLS*. Também, conforme nas outras três taxas de bits, os melhores resultados foram obtidos com o percentual de mutação de 90%, devido à alteração realizado no Método de Mutação 1. Em relação ao tempo de processamento, devido a melhor solução ter sido encontrada com o Número de Gerações igual a 500, o tempo de execução foi demasiadamente elevado, igual a 1, 37 segundos.

Através de uma análise geral das Tabelas 25, 26, 27 e 28, verifica-se que com a inserção da restrição dos valores dos *scalefactors*, para as quatro taxas de bits estudadas, o Algoritmo Genético obteve os melhores resultados para a distorção do sinal de áudio, quando comparado à técnica do codificador de referência. Além da busca do AG obter melhores resultados para a qualidade perceptual, o tempo de processamento do segundo experimento, em relação ao primeiro, foi menor. Desta forma, verifica-se o imenso ganho que o algoritmo obteve com a alteração do Método de Mutação 1, conseguindo produzir resultados melhores, em menos tempo. É importante salientar, que dentro do Método de Mutação 1, apenas os *scalefactors* tiveram a sua alteração de valores restringida. Os valores de *HCBS* permaneceram sendo alterados de maneira totalmente aleatória, atendendo a sua faixa específica de valores.

Deve-se observar que as melhores soluções encontradas pelo Algoritmo Genético no segundo experimento, ocorreram quando este estava configurado com 90% de mutação. No primeiro experimento, as melhores soluções foram encontradas com apenas 30% de mutação. Conforme a Figura 44, o percentual de mutação do algoritmo desenvolvido, define o percentual da população de indivíduos que sofrerão mutação, e também o percentual de *SFBs* do *frame* que terão os valores de seus *scalefactors* e *HCBS* alterados. Com a restrição dos possíveis valores de *scalefactors* para três unidades a partir do valor atual, foi provocada uma necessidade maior de diversificação do algoritmo, aumentando a taxa de mutação.

Desta forma, a redução da faixa dos valores dos *scalefactors*, realizada no segundo experimento, justifica este aumento demasiado de mutação, quando comparado aos dois experimentos. Na realidade, o processo de busca do AG se tornou mais eficiente, variando pouco os valores dos *scalefactors*, sobre muitos indivíduos e *SFBs*, do que variando muito os valores dos *scalefactors*, sobre poucos indivíduos e *SFBs*.

Na Tabela 29 é apresentada uma comparação entre os resultados do Algoritmo Genético e da técnica do codificador de referência, para as quatro taxas de bits. Para facilitar a análise, os valores foram representados por uma relação em dB e por um valor percentual. Desta forma, a tabela expressa a atenuação de distorção, em dB e em por cento, obtida pelo Algoritmo Genético, quando comparado com o método do codificador de referência MPEG-4 AAC.

<i>ANMR</i>	Bit rate			
	48kbps	116kbps	184kbps	250kbps
μ (dB)	-5,22	-9,74	-3,89	-4,96
μ (%)	-45,2%	-67,4%	-36,1%	-43,5%

Tabela 29 – Algoritmo Genético vs *Two Loop Search*

Através desta análise, identificam-se melhoras significativas da qualidade perceptual do sinal de áudio em todas as taxas de bits estudadas. Para 116kbps, por exemplo, a atenuação média de 9,74dB na distorção, representa uma melhora de aproximadamente três vezes, na qualidade perceptual de áudio obtida com o Algoritmo Genético, quando comparado ao método do codificador de referência MPEG-4 AAC.

7.3.1.2 Busca Tabu

Com o intuito de identificar quais parâmetros melhor configuram o algoritmo da Busca Tabu, experimentos semelhantes aos executados com o Algoritmo Genético foram realizados. Nesta etapa, o objetivo foi estudar a influência dos parâmetros *BT_Max* e tamanho da Lista Tabu, sobre os valores da FO,

representada pela qualidade perceptual, e sobre o tempo de processamento (μs). O tamanho da vizinhança foi mantido constante em 20 vizinhos, para todos os experimentos realizados. Na Tabela 30 são relacionados os valores de BT_Max utilizados neste estudo, e na Tabela 31 os valores de tamanho da Lista Tabu.

BT_Max
50
100
200
500
1000
2000

Tabela 30 – Relação dos valores de BT_Max

Lista Tabu
10
25
50
100
200

Tabela 31 – Relação dos valores de tamanho da Lista Tabu

Desta forma, para combinar todos os parâmetros, foram realizados 30 experimentos para cada uma das quatro taxas de bits: 48kbps, 116kbps, 184kbps e 250kbps. Devido ao comportamento estocástico das metaheurísticas, cada um dos experimentos foi repetido 50 vezes, de maneira que os resultados pudessem ser expressos por uma média (μ) e desvio padrão (σ). O arquivo de áudio teste de número 9 foi utilizado em todos os experimentos, e os valores da Tabela 15, obtidos com a técnica *TLS*, foram utilizados como referência para avaliar o desempenho do algoritmo da Busca Tabu. Inicialmente, na Tabela 32 são apresentados os valores obtidos com a Busca Tabu para a taxa de bits de 48kbps. Para facilitar a visualização, os melhores resultados foram destacados.

Lista Tabu	Parâmetros de Avaliação		BT_Max					
			50	100	200	500	1000	2000
10	ANMR - FO	μ	38.667	38.087	37.886	37.622	37.276	37.115
		σ	294	248	231	220	232	222
	Tempo de Processamento (μ s)	μ	31.695	66.097	127.651	311.971	624.649	1.356.884
		σ	569	1.437	8.980	19.396	6.050	18.648
25	ANMR - FO	μ	38.618	37.804	37.401	37.198	37.210	36.967
		σ	262	261	226	211	203	195
	Tempo de Processamento (μ s)	μ	32.204	66.394	146.874	333.186	668.742	1.396.558
		σ	529	878	32.465	6.875	7.036	19.602
50	ANMR - FO	μ	38.612	37.530	37.485	36.146	36.911	36.944
		σ	277	273	265	201	215	208
	Tempo de Processamento (μ s)	μ	33.251	68.828	155.472	340.514	689.932	1.400.698
		σ	669	1.091	31.598	4.915	6.985	18.566
100	ANMR - FO	μ	38.603	37.395	36.358	35.966	36.566	36.089
		σ	298	263	230	215	199	189
	Tempo de Processamento (μ s)	μ	32.809	70.140	157.412	367.452	697.333	1.415.299
		σ	711	1.066	32.690	4.988	7.005	22.896
200	ANMR - FO	μ	38.490	37.384	36.222	34.825	33.799	33.645
		σ	278	224	201	215	193	135
	Tempo de Processamento (μ s)	μ	33.635	71.068	165.238	393.995	824.920	1.735.504
		σ	692	1.212	35.851	4.938	6.977	31.211

Tabela 32 - Resultados da Busca Tabu para a taxa de 48kbps

Através da análise dos resultados, observa-se que os melhores resultados obtidos com a Busca Tabu para a taxa de 48kbps, possuem valores de FO menores do que o obtido com o método *TLS*. O melhor resultado da Busca Tabu diminui em aproximadamente 1,64 vezes o ruído de quantização sobre o sinal de áudio de teste, quando comparado ao resultado obtido com a técnica do codificador de referência.

Identifica-se que os melhores resultados foram conseguidos na região em que a Lista Tabu foi configurada com maiores valores. Como a característica da Lista Tabu é diversificar a busca, conduzindo a mesma para regiões ainda não exploradas do espaço de soluções, o aumento deste parâmetro fez aumentar a diversificação durante o processo de busca. Este aumento de diversificação em combinação com a intensificação gerada pelo método de geração de vizinhança, acabaram conduzindo a busca para boas regiões do espaço de soluções, onde as melhores soluções foram encontradas. Esta relação, entre FO e Lista Tabu, pode ser identificada também na Figura 56.

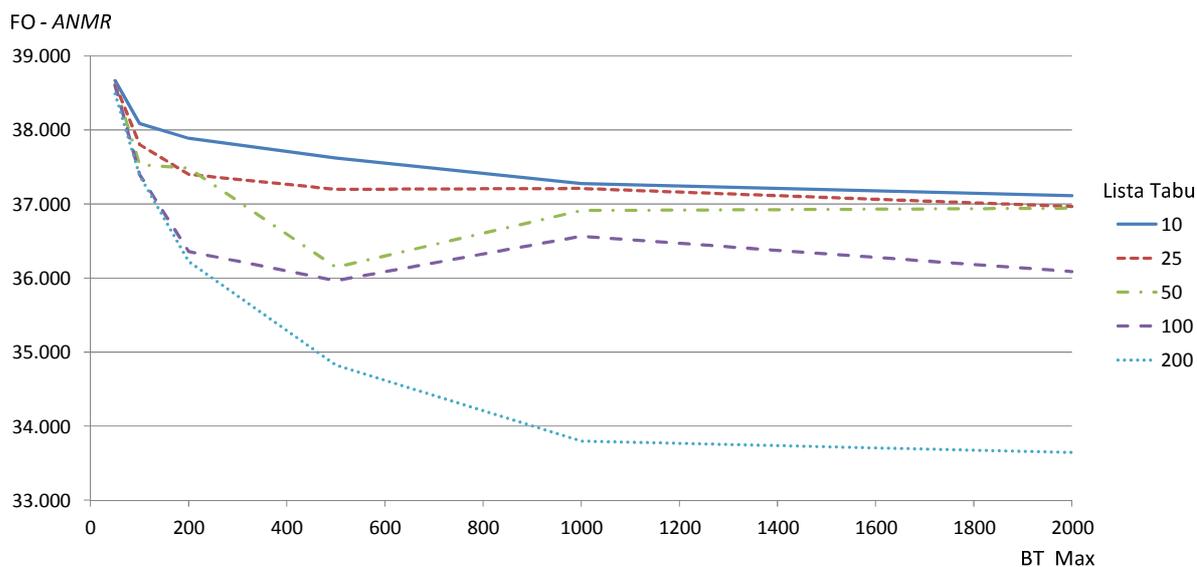


Figura 56 – Relação entre BT_Max e Lista Tabu para a taxa de 48kbps

Verifica-se também que as melhores soluções foram encontradas quando os valores de BT_Max foram maiores, ou seja, quando o tempo de busca foi estendido. Isso permitiu que a busca explorasse de maneira mais abrangente o espaço de soluções, e conseqüentemente, encontrasse melhores soluções. Em contrapartida, um aumento de BT_Max, produziu um aumento considerável do tempo de processamento. A melhor solução apresentada na Tabela 32, utilizou BT_Max igual a 2000, porém o tempo de processamento neste caso foi de 1,73 segundos. Este tempo para processar um *frame* de áudio de 43ms, é muito significativo. Com o aumento do número de BT_Max, verifica-se também, que as soluções são mais precisas, ou seja, o desvio padrão é menor.

O experimento de estudo dos parâmetros da Busca Tabu foi repetido para a taxa de 116kbps. Na Tabela 33 são apresentados os resultados obtidos, através da combinação entre os valores dos parâmetros de BT_Max e tamanho da Lista Tabu. Cada experimento foi repetido 50 vezes, e a fim de facilitar a identificação os melhores resultados foram destacados.

Lista Tabu	Parâmetros de Avaliação		BT_Max					
			50	100	200	500	1000	2000
10	ANMR - FO	μ	5.137	5.074	5.061	5.004	4.951	4.968
		σ	55	64	48	37	44	40
	Tempo de Processamento (μ s)	μ	51.390	96.441	179.650	436.363	874.471	1.658.963
		σ	770	1.881	1.568	8.373	6.794	9.123
25	ANMR - FO	μ	5.125	5.032	5.008	4.985	4.877	4.870
		σ	47	53	46	40	32	30
	Tempo de Processamento (μ s)	μ	54.760	98.609	188.595	440.621	897.406	1.789.552
		σ	2.877	884	4.525	6.442	7.960	10.265
50	ANMR - FO	μ	5.101	5.004	4.955	4.894	4.808	4.790
		σ	49	40	42	47	39	35
	Tempo de Processamento (μ s)	μ	54.992	102.856	219.145	463.646	912.419	1.802.774
		σ	1.184	1.388	2.541	4.582	7.386	9.936
100	ANMR - FO	μ	5.093	4.978	4.864	4.815	4.795	4.780
		σ	58	47	48	49	45	42
	Tempo de Processamento (μ s)	μ	55.202	109.915	214.927	485.587	1.000.458	2.256.974
		σ	1.052	2.855	3.210	4.265	10.721	12.569
200	ANMR - FO	μ	5.098	4.967	4.961	4.634	4.480	4.469
		σ	56	54	55	36	30	28
	Tempo de Processamento (μ s)	μ	56.126	111.029	221.734	573.479	1.174.822	2.469.158
		σ	1.100	1.803	2.844	11.191	23.315	35.721

Tabela 33 - Resultados da Busca Tabu para a taxa de 116kbps

Através da análise da Tabela 33, identifica-se novamente o sucesso da Busca Tabu em obter melhores soluções, do que as obtidas com o método do codificador de referência. A melhor solução encontrada pela Busca Tabu para a taxa de 116kbps, apresenta uma qualidade perceptual de áudio aproximadamente 2,57 vezes superior à obtida com o codificador de referência.

As melhores soluções foram encontradas, quando a Lista Tabu e BT_Max assumiram os maiores valores. A melhor solução da Tabela 33 foi encontrada com BT_Max igual a 2000, porém o tempo de processamento nesta condição, foi de 2,46 segundos. No entanto, é interessante observar, que a melhora neste caso, em relação à BT_Max igual a 1000, é muito pequena, apenas 0,24%. Desta forma, deve-se ponderar o tempo de processamento necessário, e a melhora inserida sobre o valor de FO, de cada configuração dos parâmetros.

Para facilitar o estudo, através da Figura 57 é possível verificar a dinâmica da relação entre FO, BT_Max e a Lista Tabu graficamente.

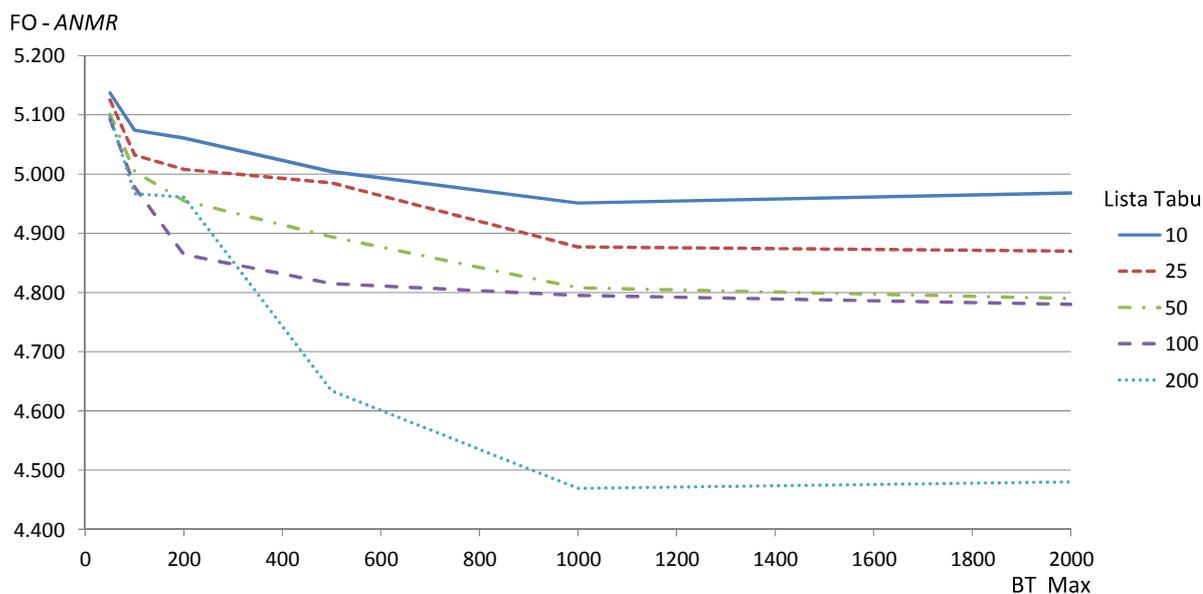


Figura 57 – Relação entre BT_Max e Lista Tabu para a taxa de 116kbps

Pode-se verificar na Figura 57 a influência da diversificação, inserida pelo aumento da Lista Tabu, sobre a melhora (diminuição) dos valores de FO. Verifica-se que com o aumento do tamanho da Lista Tabu, diversificando melhor a busca, novas regiões do espaço de soluções puderam ser alcançadas, e assim melhores soluções foram obtidas. Desta forma, diversificando a busca com a Lista Tabu e intensificando-a com a geração de vizinhança, o algoritmo da Busca Tabu foi melhor parametrizado para a taxa de 116kbps. Este comportamento verificado com a taxa de 116kbps é semelhante ao identificado para a taxa de 48kbps.

Dando sequência ao estudo dos parâmetros de configuração da Busca Tabu, o arquivo de áudio teste foi codificado utilizando a taxa de bits de 184kbps. Os experimentos foram realizados 50 vezes para cada uma das 30 combinações dos parâmetros, e assim os valores de FO e tempo de processamento foram representados por uma média (μ) e desvio padrão (σ).

A Tabela 34 apresenta os valores obtidos com o algoritmo da Busca Tabu, na solução do Problema de Codificação AAC para a taxa de 184kps. As melhores soluções foram destacadas para facilitar sua visualização.

Lista Tabu	Parâmetros de Avaliação		BT_Max					
			50	100	200	500	1000	2000
10	ANMR - FO	μ	774	766	763	763	760	758
		σ	10	12	10	8	10	8
	Tempo de Processamento (μ s)	μ	63.867	122.199	228.035	457.811	1.138.999	2.367.874
		σ	621	11.481	1.810	5.394	13.443	25.896
25	ANMR - FO	μ	773	767	764	755	757	755
		σ	14	10	10	10	10	9
	Tempo de Processamento (μ s)	μ	66.409	120.810	235.019	478.965	1.166.278	2.385.789
		σ	1.665	1.158	3.421	5.577	13.865	27.894
50	ANMR - FO	μ	773	765	760	756	755	753
		σ	14	12	12	12	12	11
	Tempo de Processamento (μ s)	μ	65.999	124.412	241.896	476.384	1.238.004	2.496.755
		σ	1.162	2.446	3.369	6.941	15.842	26.889
100	ANMR - FO	μ	773	765	759	750	751	749
		σ	13	11	11	10	10	10
	Tempo de Processamento (μ s)	μ	67.484	128.099	255.047	480.303	1.278.955	2.578.863
		σ	3.063	2.539	5.552	7.944	18.900	32.457
200	ANMR - FO	μ	773	764	755	748	742	744
		σ	14	11	11	9	8	8
	Tempo de Processamento (μ s)	μ	66.962	128.391	268.942	467.066	1.275.464	2.879.120
		σ	1.747	2.571	5.889	11.484	30.773	38.966

Tabela 34 – Resultados da Busca Tabu para a taxa de 184kbps

Para taxa de 184kbps, o algoritmo da Busca Tabu obteve novamente resultados melhores do que os alcançados pelo algoritmo *TLS* do codificador de referência. A melhor solução da Busca Tabu representou uma melhora de aproximadamente 1,46 vezes na qualidade perceptual do sinal de áudio codificado, em relação ao método de referência. Deve-se observar que os melhores resultados foram conseguidos com maiores valores de BT_Max e de tamanho da Lista Tabu. Isso significa que um maior tempo de busca e um maior tempo de proibição para que uma solução possa ser visitada novamente, melhoraram o desempenho do algoritmo.

A estratégia de diversificação, com o aumento da Lista Tabu, novamente acabou conduzindo o algoritmo da Busca Tabu para os melhores resultados. Com o aumento de BT_Max, as soluções também foram guiadas para boas regiões, porém o tempo de processamento, acabou sendo significativamente elevado, se tratando da codificação de um *frame*. Como exemplo, pode-se citar a melhor solução da

Tabela 34, que foi obtida com BT_Max igual a 1000, com um tempo de processamento de 1,27 segundos.

Outro ponto que deve ser comentado, é que devido os valores de desvio padrão das soluções em destaque da tabela serem baixos, o algoritmo demonstra precisão e convergência, na solução do Problema de Codificação AAC.

A fim de ilustrar o comportamento entre a qualidade perceptual, representada por FO, a Lista Tabu e BT_Max é apresentada a Figura 58.

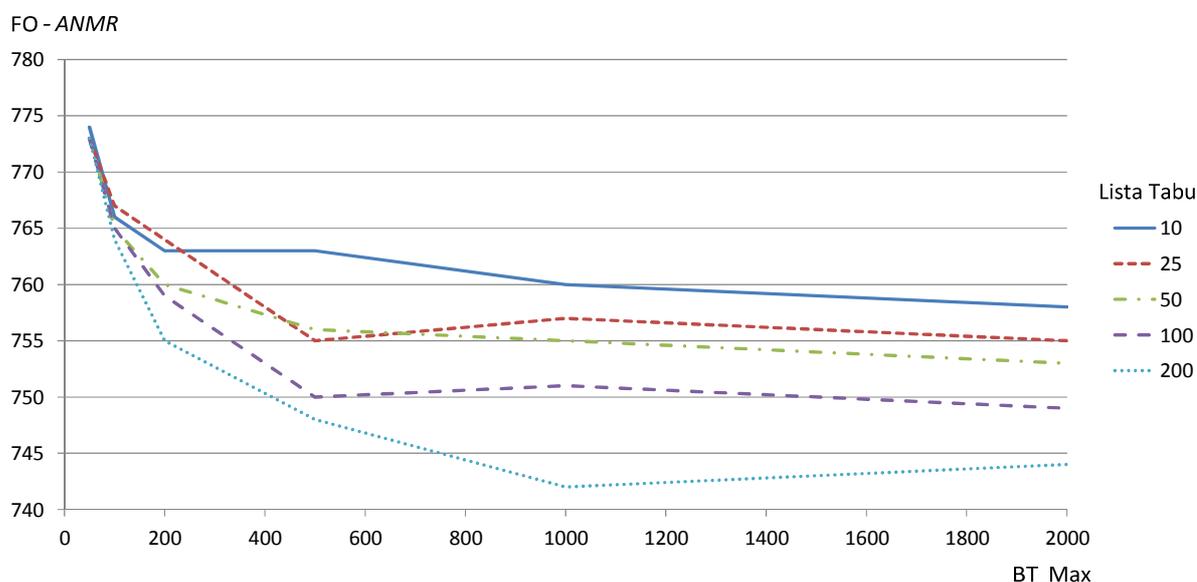


Figura 58 – Relação entre BT_Max e Lista Tabu para a taxa de 184kbps

Através desta análise gráfica é interessante observar que para valores pequenos de BT_Max, a Busca Tabu não explora com muita qualidade o espaço de soluções. Porém, à medida que BT_Max vai aumentando, tende a melhorar as soluções. Verifica-se que basicamente a partir de BT_Max igual a 500, há uma estabilização da conduta das soluções, para os diferentes tamanhos de Lista Tabu. A única exceção é para o tamanho da Lista igual a 200, que obtém uma boa melhoria com BT_Max igual a 1000.

Através desta análise da dinâmica dos valores de FO, deve-se observar que para todas as combinações da Tabela 34, foram encontrados melhores valores de qualidade perceptual de áudio, do que a obtida pela técnica do codificador de referência.

O último teste de parametrização da Busca Tabu foi realizado utilizando a taxa de 250kbps. Para a frequência de amostragem utilizada de 44,1kHz, esta é a máxima taxa de bits que se pode utilizar, segundo a normatização do codificador (ISO/IEC, 2004). Na Tabela 35 são apresentados os valores obtidos nos experimentos das 30 combinações entre os parâmetros da Busca Tabu. Ambos os valores em análise, FO e tempo de processamento, foram representados por uma média (μ) e desvio padrão (σ), devido o comportamento estocástico do algoritmo da Busca Tabu. Os melhores resultados foram destacados, para facilitar a visualização.

Lista Tabu	Parâmetros de Avaliação		BT_Max					
			50	100	200	500	1000	2000
10	ANMR - FO	μ	189	184	182	176	169	167
		σ	4	3	3	2	36	2
	Tempo de Processamento (μ s)	μ	85.102	153.904	287.545	550.359	1.527.788	3.085.441
		σ	1.106	1.769	2.770	8.911	23.499	32.988
25	ANMR - FO	μ	188	183	179	168	167	167
		σ	4	3	3	3	2	1
	Tempo de Processamento (μ s)	μ	89.783	159.530	296.581	574.953	1.587.654	3.184.613
		σ	2.382	1.781	2.286	9.005	25.602	35.974
50	ANMR - FO	μ	187	183	180	172	168	166
		σ	4	3	3	2	2	2
	Tempo de Processamento (μ s)	μ	91.295	165.749	305.687	563.679	1.659.445	3.358.947
		σ	2.614	2.043	3.461	9.735	24.985	41.787
100	ANMR - FO	μ	188	181	175	168	166	165
		σ	3	3	2	2	2	1
	Tempo de Processamento (μ s)	μ	93.106	175.940	337.224	578.661	1.786.991	3.674.511
		σ	2.125	2.128	5.195	12.447	30.057	58.433
200	ANMR - FO	μ	187	180	170	166	159	155
		σ	3	3	3	2	2	1
	Tempo de Processamento (μ s)	μ	95.088	180.425	355.963	597.300	1.850.154	3.885.817
		σ	1.635	2.986	5.887	28.499	39.157	69.834

Tabela 35 – Resultados da Busca Tabu para a taxa de 250kbps

Novamente, como também identificado para as outras três taxas de bits, o algoritmo da Busca Tabu obteve melhores resultados do que os obtidos o pela técnica do codificador de referência. A solução obtida pela Busca Tabu para 250kbps, apresentou uma melhora de 2,28 vezes em relação à solução encontrada pelo método *TLS* do codificador de referência.

Com o aumento dos valores de BT_Max e da Lista Tabu, o algoritmo obteve os melhores resultados. Este comportamento pode ser identificado também na análise gráfica da Figura 59.

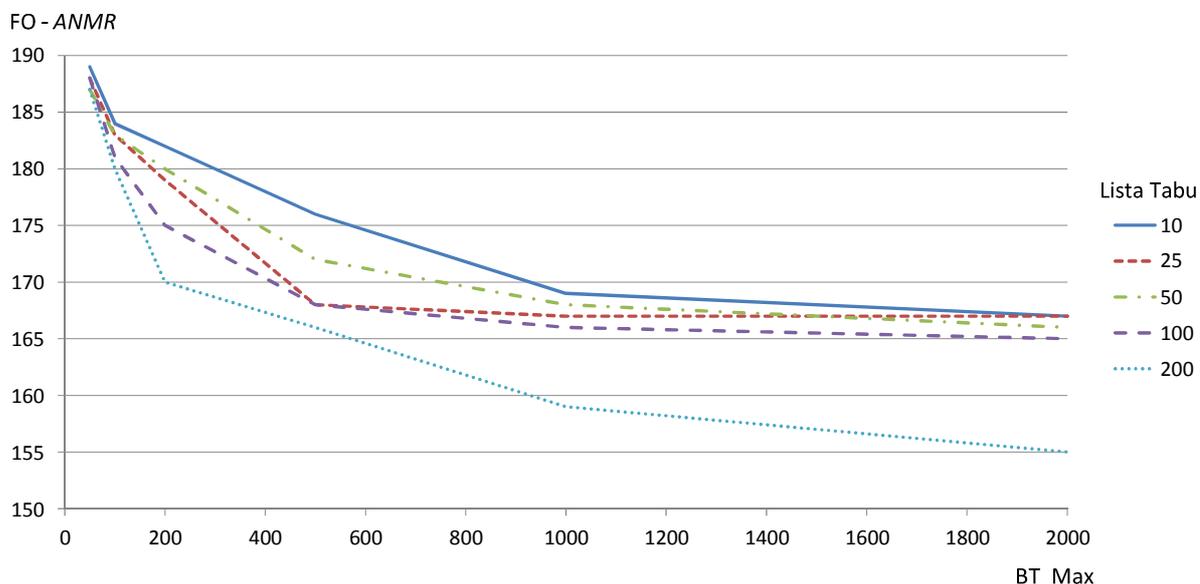


Figura 59 – Relação entre BT_Max e Lista Tabu para a taxa de 250kbps

Com o incremento da Lista Tabu, a estratégia de diversificação é colocada em evidência. Isso ocorre devido a um tempo maior de proibição aos valores que podem ser assumidos durante a busca, obriga a mesma a seguir outras trajetórias, alcançando regiões do espaço de soluções ainda não visitadas.

Pequenos valores de BT_Max não produzem soluções tão boas quanto para maiores valores do parâmetro. Porém, com o aumento de BT_Max, o tempo de processamento também cresce, devido o tempo de procura ser prolongado. Por exemplo, a melhor solução da Tabela 35, é obtida com BT_Max igual a 2000, com tempo de processamento igual a 3,88 segundos. Este tempo acaba sendo muito significativo para codificação de um *frame* de áudio. A partir de BT_Max igual a 1000, o comportamento de FO apresenta uma estabilização, para todos os diferentes tamanhos de Lista Tabu.

A fim de realizar uma comparação da atenuação da distorção de quantização perceptual (*ANMR*) sobre o sinal de áudio, obtido com a Busca Tabu, em relação à técnica *TLS*, foi elaborada a Tabela 36. Nesta, a atenuação obtida pela Busca Tabu

é representada em dB e em percentual, em relação ao método do codificador de referência.

ANMR (dB)	Bit rate			
	48kbps	116kbps	184kbps	250kbps
μ (dB)	-4,34	-8,21	-3,29	-7,17
μ (%)	-39,4%	-61,1%	-31,5%	-56,2%

Tabela 36 – Busca Tabu vs *Two Loop Search*

Para as quatro taxas de bits estudadas, o algoritmo da Busca Tabu obteve sempre um melhor resultado do que a técnica do codificador de referência. Para a taxa de 116kbps chegou a obter uma melhora sobre ANMR de 8,19 dB. Deve-se ainda ressaltar que, qualquer um dos valores de FO apresentados nas Tabelas 32, 33, 34 e 35, para qualquer combinação entre os parâmetros, é melhor do que os valores obtidos pelo método do codificador de referência, tomadas as suas respectivas taxas de bits.

Em uma análise geral do comportamento do algoritmo para as quatro taxas de bits, pôde-se identificar que diversificando a busca, com o aumento da Lista Tabu e intensificando-a com a geração de vizinhança, o algoritmo da Busca Tabu obteve os melhores resultados. Desta forma, a sua melhor parametrização para abordar o Problema de Codificação AAC pôde ser identificada.

7.3.1.3 Algoritmo Híbrido

Na terceira etapa dos experimentos de parametrização, foram estudados os parâmetros dos dois algoritmos das metaheurísticas Algoritmo Genético e Busca Tabu juntos, em uma estrutura híbrida. Desta forma, procurou-se identificar uma melhoria do valor de FO, como também melhoria em relação ao tempo de processamento.

Nesta fase foram estudados os parâmetros de Número de Gerações, Lista Tabu e BT_Max. O tamanho da vizinhança ficou estabelecido em 20 vizinhos, o tamanho da população em 20 indivíduos, a taxa de cruzamento em 90%, o processo

de seleção por torneio foi configurado com probabilidade de 80% para os mais aptos, e a taxa de mutação foi fixada em 90%, devido aos experimentos relatados na seção 7.3.1.1. A Tabela 37 apresenta os valores de BT_Max que foram utilizados nos experimentos, a Tabela 38 os valores de Lista Tabu, e na Tabela 39 os valores de Número de Gerações.

BT_Max
50
100
200

Tabela 37 – Relação dos valores de BT_Max para o Algoritmo Híbrido

Apesar de valores de BT_Max igual a 500, 1000 e 2000, terem apresentado as melhores soluções para FO durante os experimentos com a Busca Tabu, devido ao alto tempo de processamento necessário nestes casos, estes valores não foram utilizados nos experimentos do Algoritmo Híbrido. Vale ressaltar que todas as soluções encontradas com BT_Max igual a 50, 100 e 200, nos experimentos da Busca Tabu, foram de melhor qualidade do que as obtidas pela técnica do codificador de referência.

Lista Tabu
100
200

Tabela 38 – Relação dos valores de Lista Tabu para o Algoritmo Híbrido

De acordo com os experimentos da Busca Tabu, os melhores resultados foram obtidos com o tamanho de Lista Tabu igual 100 e 200. Desta forma, não foi verificada a necessidade de utilizar valores diferentes destes nos testes do Algoritmo Híbrido.

Número de Gerações
50
100
200
500

Tabela 39 – Relação do Número de Gerações para o Algoritmo Híbrido

Apesar do Número de Gerações igual a 200 e 500 terem apresentado elevados tempos de processamento, estes foram inseridos na relação dos valores dos parâmetros a serem estudados no Algoritmo Híbrido. Isso se deve ao fato do Algoritmo Genético ter obtido melhores resultados em três das quatro taxas de bits estudadas, quando comparado aos valores da Busca Tabu. Desta forma, estes valores para o Número de Gerações foram mantidos, a fim de estudar seu comportamento também dentro da estrutura híbrida dos algoritmos.

De maneira semelhante aos demais experimentos executados, o arquivo de áudio teste de número 9 foi o sinal de entrada do codificador, e assim foi codificado nas quatro taxas de bits: 48kbps, 116kbps, 184kbps e 250kbps. Para que fossem atendidas todas as combinações dos parâmetros, foram realizados 24 experimentos, que foram repetidos 50 vezes cada um. Desta forma, os valores de FO e tempo de processamento foram representados por uma média (μ) e um desvio padrão (σ). Na Tabela 40 são apresentados os resultados obtidos com o Algoritmo Híbrido para a taxa de 48kbps. Para facilitar a visualização, os melhores resultados foram destacados.

Lista Tabu	Parâmetros de Avaliação		BT_Max					
			50		100		200	
			Número de Gerações					
		50	100	50	100	50	100	
100	ANMR - FO	μ	27.329	27.392	28.436	27.374	28.251	28.598
		σ	161	181	542	146	462	601
	Tempo de Processamento (μ s)	μ	58.433	117.718	89.483	119.606	160.459	248.687
		σ	1.465	2.563	1.393	2.786	25.563	91.667
200	ANMR - FO	μ	27.327	27.340	28.428	27.382	28.478	28.474
		σ	177	181	562	180	576	623
	Tempo de Processamento (μ s)	μ	57.994	117.457	89.109	117.709	156.189	188.052
		σ	1.352	2.092	2.532	2.917	2.797	4.309
Lista Tabu	Parâmetros de Avaliação		BT_Max					
			50		100		200	
			Número de Gerações					
		200	500	200	500	200	500	
100	ANMR - FO	μ	28.545	28.903	28.591	27.906	28.575	27.856
		σ	112	153	148	146	143	105
	Tempo de Processamento (μ s)	μ	243.729	720.371	243.220	694.330	275.997	1.395.642
		σ	4.239	22.148	6.546	15.449	38.225	29.458
200	ANMR - FO	μ	28.611	29.702	28.601	27.750	28.550	27.502
		σ	81	174	141	166	149	122
	Tempo de Processamento (μ s)	μ	247.066	738.991	242.490	714.367	244.679	1.217.441
		σ	6.057	12.296	4.436	16.493	1.778	31.457

Tabela 40 – Resultados do Algoritmo Híbrido para a taxa de 48kbps

Os melhores resultados foram obtidos com uma rápida convergência do algoritmo. Com apenas 50 gerações do AG, e BT_Max igual a 50, o Algoritmo Híbrido obteve resultados melhores do que das metaheurísticas operando separadamente. Esse comportamento se deve à dinâmica do Algoritmo Híbrido. Durante seu processo de iteração, ocorrem constantes etapas de intensificação, tanto no AG (com o cruzamento) como na BT (com geração de vizinhança), conduzindo as soluções a convergirem rapidamente. Ao mesmo tempo, a Lista Tabu e a mutação provocam a diversificação das soluções, guiando a busca para novas áreas do espaço de soluções. Assim, uma vez que estes espaços são identificados, o processo de intensificação os explora de forma eficiente.

Em relação à melhor solução do Algoritmo Genético, a melhora de FO obtida pelo Algoritmo Híbrido foi de aproximadamente 10%, mantendo aproximadamente o mesmo tempo de processamento. Em relação à Busca Tabu, o Algoritmo Híbrido aumentou a qualidade perceptual em 19%, com uma redução de 93% no tempo de processamento. Estes dados demonstram a eficiência do conjunto das metaheurísticas. Com a união da diversificação do Algoritmo Genético, com a intensificação da Busca Tabu, obteve-se um novo algoritmo com um novo desempenho. A fim de estudar a dinâmica da solução dentro do Algoritmo Híbrido, na Figura 60 é apresentada a evolução da solução através do Algoritmo Genético e da Busca Tabu. No gráfico apresentado foi utilizado um *frame*, de uma das 50 execuções realizadas na codificação para a taxa de 48kbps, utilizando BT_Max e Número de Gerações igual a 50, e Lista Tabu igual a 200.

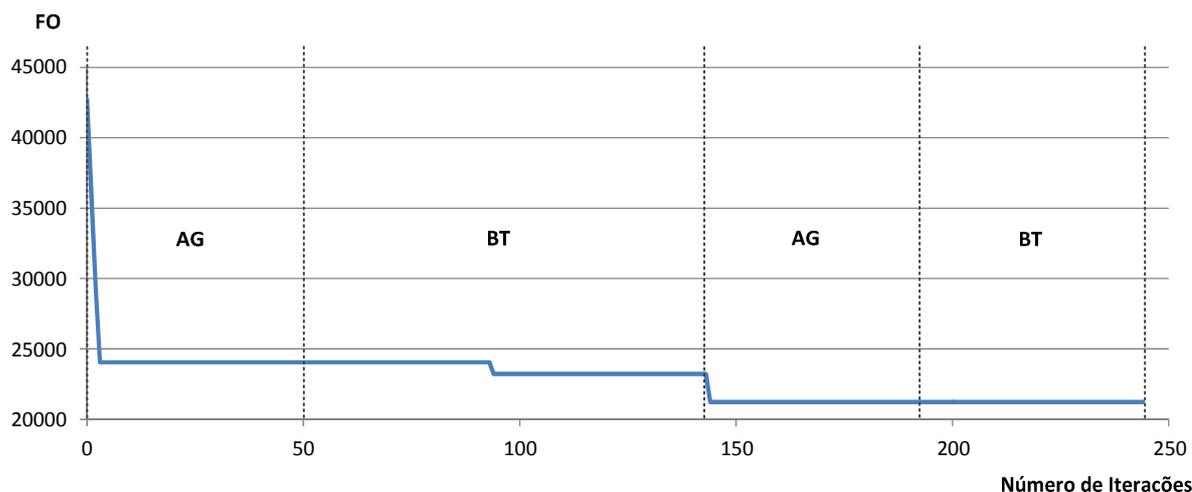


Figura 60 – Evolução do Algoritmo Híbrido em um *frame* para a taxa de 48kbps

Através da análise gráfica foi possível identificar que nas primeiras 50 iterações do AG, este conseguiu diminuir consideravelmente o valor de FO. Quando a BT foi acionada em sequência, esta também obteve sucesso, porém a redução de FO foi muito menor. A partir da nova população gerada pela BT, o AG conseguiu novamente aumentar a qualidade perceptual. Na última execução da BT, ao não ter sucesso em melhorar a sua solução inicial, o Algoritmo Híbrido foi encerrado.

O estudo de parametrização do Algoritmo Híbrido, também foi realizado para a taxa de 116kbps. Desta forma, a Tabela 41 apresenta os resultados obtidos nos 24 experimentos que combinam os parâmetros das metaheurísticas estudadas. Os valores de FO e tempo de processamento foram representados por uma média (μ) e um desvio padrão (σ), e foram realizadas 50 execuções de cada combinação. Os melhores resultados foram destacados para facilitar a visualização.

Lista Tabu	Parâmetros de Avaliação		BT_Max					
			50		100		200	
			Número de Gerações					
			50	100	50	100	50	100
100	ANMR - FO	μ	3.255	3.257	3.213	3.257	3.192	3.212
		σ	89	62	33	57	29	33
	Tempo de Processamento (μ s)	μ	75.717	158.581	120.933	161.495	208.428	260.129
		σ	1.626	2.919	2.384	3.378	3.343	23.543
200	ANMR - FO	μ	3.258	3.252	3.211	3.254	3.218	3.220
		σ	72	55	34	55	29	31
	Tempo de Processamento (μ s)	μ	75.175	161.526	120.085	158.540	210.002	257.541
		σ	1.405	4.103	2.913	3.571	3.437	11.780
Lista Tabu	Parâmetros de Avaliação		BT_Max					
			50		100		200	
			Número de Gerações					
			200	500	200	500	200	500
100	ANMR - FO	μ	3.320	3.325	3.319	3.427	3.337	3.220
		σ	40	27	36	24	51	22
	Tempo de Processamento (μ s)	μ	327.470	927.751	327.378	894.062	326.645	942.113
		σ	4.267	13.152	4.882	12.448	7.239	17.855
200	ANMR - FO	μ	3.351	3.554	3.345	3.561	3.342	3.228
		σ	60	29	51	32	28	29
	Tempo de Processamento (μ s)	μ	324.382	952.235	326.522	924.551	327.800	937.274
		σ	4.410	14.821	5.193	11.424	5.199	14.720

Tabela 41 – Resultados do Algoritmo Híbrido para a taxa de 116kbps

Os melhores resultados que foram encontrados pelo Algoritmo Híbrido são melhores que os resultados das metaheurísticas operando isoladamente. Em relação ao melhor resultado obtido pela Busca Tabu, o Algoritmo Híbrido obteve uma melhora na FO de aproximadamente 29%, com uma redução no tempo computacional de 91%. Já em relação ao Algoritmo Genético, o Algoritmo Híbrido reduziu o valor de FO em aproximadamente 15%, porém para isso, o tempo computacional foi incrementado em 30%.

As melhores combinações entre os parâmetros foram as que utilizaram um Número de Gerações do AG igual a 50, com valores de BT_Max igual a 100 e 200. Isso demonstra que o AG com apenas 50 iterações, foi capaz de identificar áreas promissoras do espaço de soluções, para que a BT pudesse intensificar a busca nestas regiões. Esta dinâmica também pode ser verificada através do gráfico de evolução das soluções através das metaheurísticas apresentada na Figura 61. Este gráfico representa um *frame* de uma das 50 execuções realizadas para a taxa de

116kbps, utilizando Número de Gerações igual a 50, BT_Max igual a 200 e Lista Tabu igual a 100.

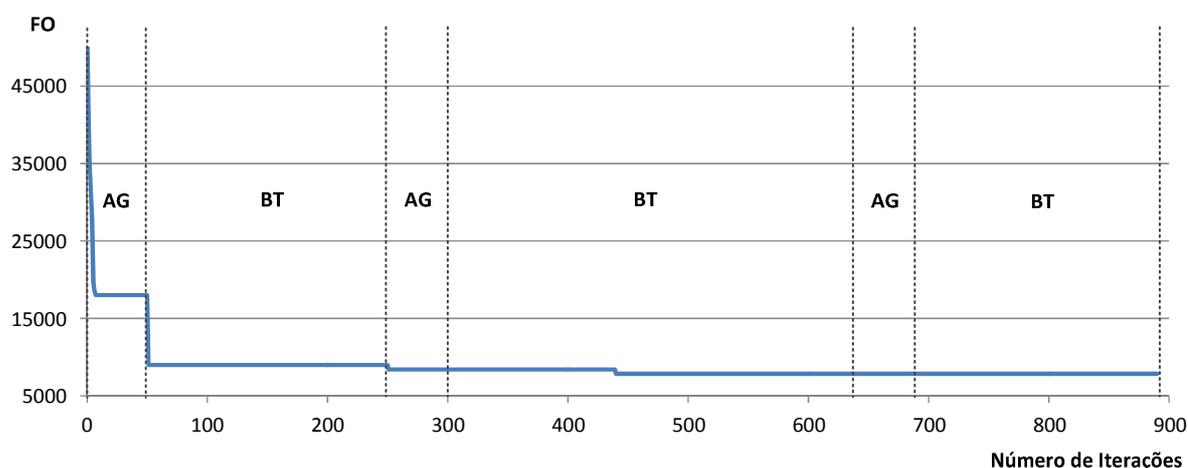


Figura 61 – Evolução do Algoritmo Híbrido em um *frame* para a taxa de 116kbps

Ao analisar a Figura 61, verifica-se que nas primeiras 50 iterações o AG rapidamente, reduz abruptamente o valor de FO. O mesmo ocorre no início da execução da BT, obtendo uma boa redução do valor de FO. Após este ponto, as reduções do valor de FO, obtidas pelas metaheurísticas, foram cada vez menores. Na última iteração do AG e BT, ambas não conseguiram melhorar a solução da BT encontrada na iteração 440. Assim depois de 890 iterações, o Algoritmo Híbrido foi encerrado.

Para a taxa de 184kbps, os experimentos para as 24 combinações dos parâmetros das metaheurísticas foram repetidos. Após terem sido realizadas 50 codificações de áudio para cada combinação, os valores de FO e tempo de processamento foram representados por uma média (μ) e um desvio padrão (σ). A Tabela 42 apresenta os resultados obtidos através dos experimentos, sendo que os melhores foram destacados.

Lista Tabu	Parâmetros de Avaliação		BT_Max					
			50		100		200	
			Número de Gerações					
			50	100	50	100	50	100
100	ANMR - FO	μ	595	595	556	588	551	554
		σ	99	66	10	55	7	7
	Tempo de Processamento (μ s)	μ	84.627	198.071	147.876	200.218	272.269	326.214
		σ	1.411	4.373	3.013	5.112	39.495	13.606
200	ANMR - FO	μ	588	581	557	593	554	555
		σ	67	46	10	57	7	8
	Tempo de Processamento (μ s)	μ	84.774	198.029	144.943	199.381	266.061	321.037
		σ	1.604	4.810	2.409	3.642	15.111	5.643
Lista Tabu	Parâmetros de Avaliação		BT_Max					
			50		100		200	
			Número de Gerações					
			200	500	200	500	200	500
100	ANMR - FO	μ	633	603	643	659	627	597
		σ	46	18	63	25	69	13
	Tempo de Processamento (μ s)	μ	415.739	1.155.743	412.156	1.058.116	414.829	1.205.193
		σ	8.521	10.757	7.172	15.402	3.944	11.902
200	ANMR - FO	μ	616	675	629	666	598	586
		σ	39	35	85	31	49	15
	Tempo de Processamento (μ s)	μ	415.190	1.188.454	414.468	1.167.395	415.825	1.169.754
		σ	8.205	16.348	5.974	14.225	7.457	11.417

Tabela 42 – Resultados do Algoritmo Híbrido para a taxa de 184kbps

Os resultados obtidos com o Algoritmo Híbrido para a taxa de 184kbps também apresentaram resultados melhores quando comparados aos resultados das metaheurísticas operando de forma separada. Em relação às soluções encontradas pela Busca Tabu, o Algoritmo Híbrido melhorou a qualidade perceptual em 25%, reduzindo o tempo computacional necessário em aproximadamente 78%. Em relação ao Algoritmo Genético, o Algoritmo Híbrido também obteve uma boa melhora, com uma redução de 20% no valor de FO, porém o tempo computacional foi incrementado demasiadamente em aproximadamente 220%.

Novamente as melhores soluções foram encontradas com o Número de Gerações do AG igual a 50. São nestas primeiras gerações, que o AG identifica as regiões mais promissoras do espaço de soluções, e repassa a tarefa de intensificação para a Busca Tabu, que explora a área de forma eficiente em busca de boas soluções. Este comportamento também pôde ser identificado na Figura 62. Nesta é possível verificar graficamente a evolução da solução de um *frame*, de uma das 50 execuções do Algoritmo Híbrido para taxa de 184kbps, utilizando como

parâmetros das metaheurísticas: BT_Max igual 200, Lista Tabu igual a 100 e Número de Gerações igual a 50. Sobre a Figura 62, deve-se ainda ressaltar que o valor de FO para a iteração 0 é igual a 55193. Porém, este ponto não é representado no gráfico, apenas para facilitar a visualização do mesmo, já que existe uma discrepância muito grande entre o valor inicial e final de FO.

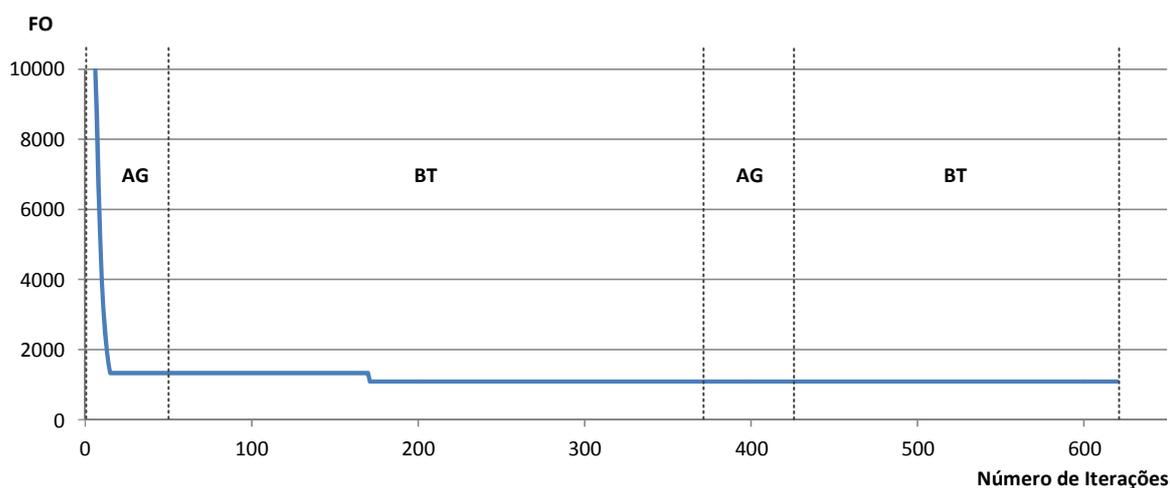


Figura 62 – Evolução do Algoritmo Híbrido em um *frame* para a taxa de 184kbps

É possível identificar através da Figura 62, que nas primeiras iterações do AG, o valor de FO foi drasticamente diminuído, e depois se percebe uma estabilização do seu valor. A BT conseguiu melhorar a solução na iteração de número 112, porém após este ponto, ambas metaheurísticas não obtiveram o mesmo sucesso, e assim o Algoritmo Híbrido foi encerrado com 620 iterações.

A fim de estudar a dinâmica entre os parâmetros das duas metaheurísticas em uma única estrutura híbrida para a taxa de 250kbps, os experimentos contemplando as 24 combinações dos parâmetros, foram repetidos. Desta forma, os resultados são apresentados na Tabela 43, sendo representados os valores de FO e tempo de processamento por uma média (μ) e um desvio padrão (σ).

Lista Tabu	Parâmetros de Avaliação		BT_Max					
			50		100		200	
			Número de Gerações					
			50	100	50	100	50	100
100	ANMR - FO	μ	184	164	104	164	102	103
		σ	94	77	2	39	2	2
	Tempo de Processamento (μ s)	μ	95.178	232.061	181.143	234.188	333.989	437.854
		σ	4.110	4.429	4.368	4.186	13.290	52.160
200	ANMR - FO	μ	178	169	104	173	102	102
		σ	96	61	2	59	2	1
	Tempo de Processamento (μ s)	μ	94.244	231.641	178.330	234.471	344.169	403.683
		σ	1.879	3.447	4.116	3.930	26.326	8.245
Lista Tabu	Parâmetros de Avaliação		BT_Max					
			50		100		200	
			Número de Gerações					
			200	500	200	500	200	500
100	ANMR - FO	μ	177	183	181	195	198	174
		σ	36	22	43	24	33	19
	Tempo de Processamento (μ s)	μ	497.047	1.346.135	492.856	1.355.801	493.513	1.398.225
		σ	4.419	13.636	5.892	12.912	3.648	30.224
200	ANMR - FO	μ	209	203	182	193	202	139
		σ	62	45	40	28	46	17
	Tempo de Processamento (μ s)	μ	491.553	1.387.813	497.562	1.392.878	494.952	1.355.509
		σ	6.320	8.178	4.809	30.952	6.928	28.298

Tabela 43 – Resultados do Algoritmo Híbrido para a taxa de 250kbps

Através de uma análise, identifica-se que novamente, conforme observado para as outras três taxas de bits, os resultados obtidos com o Algoritmo Híbrido foram melhores do que os encontrados pela Busca Tabu e pelo Algoritmo Genético operando sozinhos. Em relação ao melhor valor obtido para FO com a Busca Tabu, o Algoritmo Híbrido diminuiu o valor de FO em aproximadamente 34%, reduzindo o tempo de processamento em 91%. Em comparação com o melhor valor obtido do Algoritmo Genético para a taxa de 250kbps, o Algoritmo Híbrido reduziu o valor de FO em 49%, e o tempo computacional em 75%.

Pode-se observar uma forte convergência dos resultados, quando o valor de BT_Max é igual a 200 para a parte superior da Tabela 43. Os valores encontrados são muito próximos e com baixos valores de desvio padrão. Isso ocorre, pois devido o aumento da taxa de bits ter provocado um aumento no espaço de soluções, o processo de busca acabou sendo melhor conduzido, com um aumento da estratégia de intensificação. Esta característica foi provida pelo aumento do parâmetro BT_Max.

A fim de estudar a evolução da solução dentro das metaheurísticas, a Figura 63, apresenta um gráfico da dinâmica interna do Algoritmo Híbrido. Nesta é apresentada a evolução da solução de um *frame*, de uma das 50 execuções do Algoritmo Híbrido para a taxa de 250kbps, com os parâmetros definidos como: Número de Gerações igual a 50, BT_Max igual a 200 e Lista Tabu igual a 100. O valor de FO no início da busca para iteração igual a 0, foi igual a 32267. Porém, devido à grande diferença entre os valores final e inicial de FO, este valor não foi representado na Figura 63. Esta é apenas uma medida para facilitar a identificação da dinâmica interna da solução do Algoritmo Híbrido.

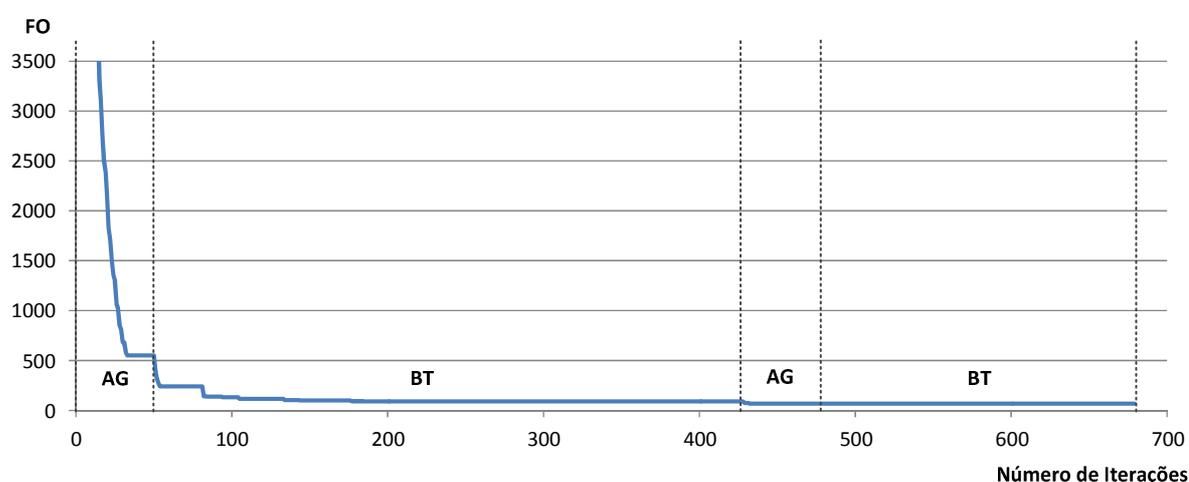


Figura 63 – Evolução do Algoritmo Híbrido em um *frame* para a taxa de 250kbps

Através da Figura 63, pode-se observar que o AG em suas primeiras 50 gerações, reduziu com muita eficiência o valor de FO. Na sequência, a BT permaneceu na busca durante aproximadamente 380 iterações. Nesta etapa é que ocorreu a intensificação da busca através da BT, e conforme o algoritmo foi encontrando melhores soluções, ele se manteve nesta condição. Ao atingir o valor de BT_Max, a BT repassou a nova população para o AG, que diversificou-a, e obteve melhores resultados. As últimas iterações da BT não obtiveram sucesso, e assim o Algoritmo Híbrido foi encerrado. Na Tabela 44 é realizada uma comparação, em dB, dos valores médios de *ANMR* obtidos com o Algoritmo Híbrido, em relação às demais técnicas, *Two Loop Search*, Algoritmo Genético e Busca Tabu

<i>ANMR - μ (dB)</i>	Bit rate			
	48kbps	116kbps	184kbps	250kbps
<i>TLS</i>	-6,15	-11,13	-5,88	-10,81
<i>AG</i>	-0,93	-1,40	-1,99	-5,85
<i>BT</i>	-1,81	-2,92	-2,59	-3,63

Tabela 44 – Comparação entre o Algoritmo Híbrido e demais métodos

. Através da Tabela 44 é possível identificar que o Algoritmo Híbrido obteve os melhores resultados para *ANMR*, quando comparado com os demais métodos, *Two Loop Search*, Algoritmo Genético e Busca Tabu.

Para que a dinâmica da estrutura do Algoritmo Híbrido pudesse ser totalmente estudada, foram realizados alguns experimentos utilizando uma configuração diferente entre as metaheurísticas: primeiro a Busca Tabu e na sequência o Algoritmo Genético. A ideia não foi alterar características dos algoritmos, mas apenas realizar a inversão da ordem de aplicação, e estudar os resultados obtidos através de experimentos. Desta forma, os experimentos foram realizados para as quatro taxas de bits, com o arquivo de áudio teste de número 9 como entrada do codificador, repetindo a execução de cada combinação entre os parâmetros por 50 vezes, de maneira que os valores de FO e tempo de processamento foram representados por uma média (μ) e um desvio padrão (σ). Nestes experimentos os valores de Número de Gerações foram restringidos apenas para 50 e 100, diferentemente dos experimentos anteriores que contemplaram também os valores 200 e 500. Esta medida foi estabelecida em função dos resultados para estes valores não terem se destacado entre os demais. Desta forma, na Tabela 45 são apresentados os resultados do Algoritmo Híbrido Invertido.

Lista Tabu	Parâmetros de Avaliação		BT_Max					
			50		100		200	
			Número de Gerações					
			50	100	50	100	50	100
100	ANMR - FO	μ	38.323	38.267	37.226	37.282	36.134	36.122
		σ	179	226	86	317	280	96
	Tempo de Processamento (μ s)	μ	94.850	167.028	135.414	209.606	217.446	285.171
		σ	2.545	2.554	2.954	1.833	1.319	4.232
200	ANMR - FO	μ	38.192	38.420	37.138	37.153	36.231	36.057
		σ	189	115	438	235	153	116
	Tempo de Processamento (μ s)	μ	94.036	171.029	135.794	206.693	216.487	291.018
		σ	1.011	1.479	2.319	3.672	977	2.741

Tabela 45 – Resultados do Algoritmo Híbrido Invertido para a taxa de 48kbps

Os resultados obtidos com o Algoritmo Híbrido Invertido para a taxa de 48kbps, não foram melhores do que os encontrados pelo Algoritmo Híbrido. Através da Tabela 45, pode-se perceber uma mudança na forma em que o algoritmo executou a busca. Com o Número de Gerações igual a 100, identifica-se a necessidade dos operadores genéticos de mutação e cruzamento, operarem por mais tempo durante busca. Assim, através da diversificação, a busca foi guiada para regiões promissoras no espaço de soluções, as quais foram intensificadas pela BT. Com valor de BT_Max igual a 200, o algoritmo investiu um bom tempo na intensificação das regiões. E este é outro diferencial que deve ser apontado. Com mais tempo de diversificação, e mais tempo de intensificação, o tempo computacional utilizado pelo Algoritmo Híbrido Invertido acabou sendo consideravelmente superior ao tempo de processamento obtido com o Algoritmo Híbrido.

Na Tabela 46 são apresentados os resultados dos experimentos do Algoritmo Híbrido Invertido para a taxa de 116kbps. De maneira semelhante aos demais experimentos, cada uma das 12 combinações dos parâmetros da metaheurísticas foi executada 50 vezes, de maneira que os valores de FO e tempo de processamento foram representados por uma média (μ) e um desvio padrão (σ).

Lista Tabu	Parâmetros de Avaliação		BT_Max					
			50		100		200	
			Número de Gerações					
			50	100	50	100	50	100
100	ANMR - FO	μ	4.560	4.649	4.600	4.530	4.518	4.514
		σ	55	37	62	67	9	26
	Tempo de Processamento (μ s)	μ	192.959	326.437	258.220	393.013	396.471	533.414
		σ	3.269	5.560	4.047	8.025	1.052	9.806
200	ANMR - FO	μ	4.687	4.660	4.618	4.573	4.466	4.483
		σ	35	41	48	13	20	48
	Tempo de Processamento (μ s)	μ	185.998	320.258	255.922	389.220	408.979	532.308
		σ	4.084	10.144	3.250	6.298	1.014	10.284

Tabela 46 – Resultados do Algoritmo Híbrido Invertido para a taxa de 116kbps

A qualidade das soluções obtidas com o Algoritmo Híbrido Invertido não superou a qualidade das soluções encontradas pelo Algoritmo Híbrido, apesar de todas serem melhores do que a solução do codificador de referência. Com o valor da Lista Tabu igual a 200, verifica-se a necessidade de alta diversificação no início da busca, a fim de encontrar áreas promissoras no espaço de soluções. Esta mesma característica foi identificada no Algoritmo Híbrido, quando esta tarefa de diversificação era realizada pelo AG nas primeiras gerações. Nota-se também, que as melhores soluções encontradas com o Algoritmo Híbrido Invertido, demandaram um maior tempo computacional, quando comparadas ao tempo necessário para encontrar as melhores soluções do Algoritmo Híbrido.

Os resultados obtidos com o Algoritmo Híbrido Invertido para a taxa de 184kbps são apresentados na Tabela 47. Os melhores resultados foram destacados para facilitar a identificação.

Lista Tabu	Parâmetros de Avaliação		BT_Max					
			50		100		200	
			Número de Gerações					
			50	100	50	100	50	100
100	ANMR - FO	μ	659	670	658	651	645	655
		σ	23	18	21	6	7	21
	Tempo de Processamento (μ s)	μ	253.010	419.628	345.486	519.627	525.951	919.097
		σ	2.180	12.756	2.648	7.344	9.207	32.705
200	ANMR - FO	μ	653	665	659	659	625	649
		σ	9	15	24	27	7	20
	Tempo de Processamento (μ s)	μ	256.329	419.830	336.945	516.823	561.978	948.820
		σ	3.070	5.498	5.306	5.834	5.764	35.361

Tabela 47 – Resultados do Algoritmo Híbrido Invertido para a taxa de 184kbps

Através da análise da Tabela 47, pôde-se verificar que as soluções encontradas pelo Algoritmo Híbrido Invertido para a taxa de 184kbps, foram de menor qualidade, quando comparadas às soluções do Algoritmo Híbrido. As melhores soluções foram obtidas com valor de Número de Gerações igual a 50 e BT_Max igual a 200. Isso demonstra que neste caso, a estratégia de intensificação da BT, através da geração de vizinhança, se tornou mais significativa durante a busca, do que a intensificação e diversificação propostas pelo AG. Verifica-se ainda que a diversificação inserida pela Lista Tabu igual a 200, produziu o melhor resultado para esta taxa de bits.

Por fim, os resultados do Algoritmo Híbrido Invertido para a taxa de 250kbps são apresentados na Tabela 48. Os experimentos foram executados 50 vezes, de maneira que os valores de FO e tempo de processamento foram representados por uma média (μ) e um desvio padrão (σ).

Lista Tabu	Parâmetros de Avaliação		BT_Max					
			50		100		200	
			Número de Gerações					
			50	100	50	100	50	100
100	ANMR - FO	μ	149	151	147	146	142	144
		σ	2	3	2	1	2	1
	Tempo de Processamento (μ s)	μ	334.643	556.867	475.957	660.699	688.682	882.000
		σ	1.513	9.267	10.285	233	11.109	26.212
200	ANMR - FO	μ	148	151	146	147	142	143
		σ	2	3	3	1	2	2
	Tempo de Processamento (μ s)	μ	337.610	553.562	452.850	663.362	697.571	1.093.708
		σ	3.758	4.293	7.865	4.718	8.754	27.145

Tabela 48 – Resultados do Algoritmo Híbrido Invertido para a taxa de 250kbps

Neste último estudo verifica-se novamente que a estrutura proposta no Algoritmo Híbrido Invertido não se mostrou tão eficiente quanto ao Algoritmo Híbrido. A qualidade dos resultados foi novamente inferior ao Algoritmo Híbrido. Neste caso de 250kbps, identifica-se que a diversificação proposta pela Lista Tabu, tem pouca influência sobre os resultados. O algoritmo obteve os melhores resultados com o valor de BT_Max igual a 200 e o Número de Gerações igual a 50. Desta forma, a BT intensificou a busca, formando uma população de soluções de elite, que o AG diversificou, procurando novas regiões promissoras no espaço de soluções.

Ao término dos experimentos com o Algoritmo Híbrido Invertido, que operou com a estrutura Busca Tabu mais Algoritmo Genético, pode-se comentar que para nenhuma das taxas de bits estudadas, os resultados foram de melhor qualidade do que os obtidos com o Algoritmo Híbrido, que utilizou a estrutura Algoritmo Genético mais Busca Tabu. Porém, deve-se ressaltar que todas as soluções encontradas, pelo Algoritmo Híbrido Invertido, com as diferentes combinações entre os parâmetros das metaheurísticas, apresentaram uma qualidade perceptual superior às obtidas com a técnica *TLS* do codificador de referência MPEG-4 AAC, apresentadas na Tabela 15. Na Tabela 49 é apresentada uma comparação, em dB, dos valores médios de *ANMR* obtidos com o Algoritmo Híbrido Invertido, em relação aos demais métodos, *Two Loop Search*, Algoritmo Genético e Busca Tabu. Através desta análise, identifica-se que o Algoritmo Híbrido Invertido, obtém os melhores resultados para *ANMR*, sempre que comparado com a *TLS*. Porém em relação ao AG e à BT, para algumas taxas de bits, o Algoritmo Híbrido Invertido não consegue melhorar as soluções já encontradas.

<i>ANMR</i> - μ (dB)	Bit rate			
	48kbps	116kbps	184kbps	250kbps
<i>TLS</i>	-3,74	-8,22	-4,78	-7,93
AG	1,48	1,52	-0,90	-2,97
BT	-0,60	0,01	1,49	0,76

Tabela 49 - Comparação entre o Algoritmo Híbrido Invertido e demais métodos

7.3.2 Resultados

Para que se pudesse avaliar a estrutura de otimização desenvolvida para o codificador MPEG-4 AAC, quanto à qualidade perceptual, foram realizados alguns experimentos de aplicação, utilizando a biblioteca de arquivos de áudio teste. Desta forma, foram realizados dois ensaios:

- **Ensaio 1:**

Inicialmente, os dez sons do conjunto de arquivos de áudio teste no formato WAV foram codificados utilizando o codificador MPEG-4 AAC de referência, utilizando as taxas de 48kbps, 116kbps, 184kbps e 250kbps, gerando arquivos de saída no formato AAC. Neste ensaio, o codificador utilizou a sua própria estrutura de controle de taxa de bits e distorção, através do método *TLS*. Assim, foram obtidos os valores de FO, tamanho de arquivo e tempo de processamento, para cada arquivo de áudio teste;

- **Ensaio 2:**

Neste ensaio, os experimentos foram realizados utilizando a estrutura de otimização baseada em metaheurísticas desenvolvida, no lugar do método *TLS*. Assim, para cada arquivo de áudio teste, foram gerados novos arquivos no formato AAC, para as taxas de 48kbps, 116kbps, 184kbps e 250kbps. Para todos estes experimentos, a estrutura de otimização foi utilizada em três configurações: Busca Tabu, Algoritmo Genético e Algoritmo Híbrido. Desta forma, os valores de FO, tamanho de arquivo e tempo de processamento, foram armazenados para cada um dos dez arquivos de áudio teste, nas três configurações. Devido ao comportamento estocástico das metaheurísticas, o experimento de codificação de cada arquivo WAV, foi repetido por 50 vezes, para que se pudesse obter um valor médio. Pelo

Teorema do Limite Central, experimentos com uma grande quantidade de amostras, tendem a ser representados por uma distribuição normal com média μ (GRIMMETT e STIRZAKER, 2001). Pela Lei dos Grandes Números, conforme uma experiência é repetida várias vezes, a probabilidade média observada aproxima-se da probabilidade real do experimento (LAW e KELTON, 2000). Desta forma, os valores de FO, tamanho de arquivo e tempo de processamento, puderam estar adequados, para efetivamente realizar uma comparação com os demais valores providos da estrutura de referência do codificador, obtidos no Ensaio 1.

Nestes ensaios, foi avaliado o tamanho de cada arquivo gerado com o processo de codificação. Esta é uma análise interessante, para que se possa identificar se o processo de codificação obteve sucesso em atingir a taxa de bits pré-determinada, sem excedê-la demasiadamente. O tamanho do arquivo é definido em função do tempo de duração de áudio do arquivo (segundos), e da taxa de bits (bits por segundo). Na Tabela 50 são apresentados os valores de tamanho de arquivo, em bytes, para os dez sons da biblioteca de áudio teste, para uma codificação nas quatro taxas de bits estudadas: 48kbps, 116kbps, 184kbps e 250kbps.

Tamanho dos arquivos (Bytes)					
Index do Arquivo	Duração (s)	Taxa de bits			
		48kbps	116kbps	184kbps	250kbps
1	8,3	53.120	120.350	190.900	259.375
2	15,3	91.800	221.850	351.900	478.125
3	9,8	58.800	142.100	225.400	306.250
4	10,25	61.500	148.625	235.750	320.313
5	3,25	19.500	47.125	74.750	101.563
6	10,2	61.200	147.900	234.600	318.750
7	10,8	64.800	156.600	248.400	337.500
8	15,7	94.200	227.650	361.100	490.625
9	15,15	90.900	219.675	348.450	473.438
10	8,25	49.500	119.625	189.750	257.813

Tabela 50 – Tamanho dos arquivos de áudio teste para cada taxa de bits

Para os ensaios com os dez arquivos de áudio teste, as metaheurísticas tiveram seus parâmetros configurados com valores baseados nos experimentos de parametrização realizados para as três configurações: Busca Tabu, Algoritmo Genético e Algoritmo Híbrido. Desta forma, a Busca Tabu foi configurada com

BT_Max igual 500 e Lista Tabu igual a 200. O Algoritmo Genético foi configurado com taxa de mutação de 90%, e Número de Gerações igual a 100. Por fim, o Algoritmo Híbrido, foi configurado com BT_Max igual a 50, Lista Tabu igual a 200 e Número de Gerações igual a 50 para a taxa de 48kbps; BT_Max igual a 200, Lista Tabu igual a 100 e Número de Gerações igual a 50 para as taxas de 116kbps, 184kbps e 250kbps. Na Tabela 51 são apresentados os valores de FO, tempo de processamento e tamanho de arquivo, para os dez sons de áudio teste, codificados com os quatro métodos: *TLS*, Busca Tabu, Algoritmo Genético e Algoritmo Híbrido. O melhor resultado para cada arquivo de áudio foi destacado para facilitar a visualização.

Método de Solução	Parâmetros de Avaliação	Índice	1	2	3	4	5	6	7	8	9	10
		Tipo de som	Bateria	Blues	Contra baixo	Coral	<i>Crash China</i>	Cravo	Narração	Orquestra	Rock	Sax
Codificador de Referência <i>TLS</i>	ANMR - FO		3.467	9.541	275	1.519	9.306	5.131	410	934	55.476	1.401
	Tamanho do arquivo		54.217	92.842	58.173	62.702	19.682	62.143	68.369	95.139	93.843	50.783
	Tempo de Processamento		33.945	33.437	32.274	31.913	32.974	33.853	32.852	33.606	32.410	33.591
Busca Tabu	ANMR - FO	μ	2.982	4.758	151	1.832	8.023	3.486	122	983	35.966	1.432
		σ	59	45	7	23	198	48	2	12	215	22
	Tamanho do arquivo	μ	52.897	91.344	58.396	60.818	19.979	59.457	66.909	93.430	90.004	49.802
		σ	140	196	93	104	85	173	138	135	262	99
	Tempo de Processamento	μ	347.378	293.491	269.605	253.641	295.811	331.071	351.575	270.050	367.452	297.571
		σ	4.907	5.676	6.543	3.820	11.327	6.989	9.753	4.110	4.988	4.547
Algoritmo Genético	ANMR - FO	μ	2.597	4.055	170	1.620	6.964	3.043	154	892	30.522	1.343
		σ	57	66	19	120	447	230	30	36	182	80
	Tamanho do arquivo	μ	50.446	88.907	55.879	58.933	18.637	58.417	62.626	90.701	87.566	48.036
		σ	322	277	391	293	480	698	731	243	242	430
	Tempo de Processamento	μ	140.491	120.136	91.719	101.572	121.983	128.010	119.834	106.197	117.746	121.830
		σ	3.530	2.714	4.300	5.020	7.949	4.896	3.577	3.550	3.082	2.659
Algoritmo Híbrido	ANMR - FO	μ	2.485	3.778	143	1.356	6.197	2.708	117	740	27.327	1.130
		σ	93	86	37	69	172	83	17	72	177	76
	Tamanho do arquivo	μ	51.674	91.747	58.089	61.337	19.164	60.969	64.988	94.149	88.942	49.952
		σ	449	405	598	211	773	315	1.012	529	295	385
	Tempo de Processamento	μ	58.188	57.002	35.813	47.481	48.319	61.320	42.060	47.929	57.994	54.423
		σ	1.510	1.102	1.055	1.662	2.593	1.481	2.440	1.289	1.352	1.439

Tabela 51 – Resultados da codificação para a taxa de 48kbps

Através da Tabela 51 observa-se para que todos os dez sons da biblioteca de áudio, os resultados de melhor qualidade foram encontrados pelo Algoritmo Híbrido. Com valores de desvio padrão baixos, demonstram que as soluções convergem e possuem uma boa precisão, o que confere confiabilidade ao Algoritmo Híbrido em encontrar boas soluções para o problema estudado. Além da qualidade perceptual

do Algoritmo Híbrido ser superior ao método *TLS*, todos os arquivos foram codificados com uma quantidade de bits menor, ou seja, com um tamanho de arquivo menor. Isso demonstra uma grande eficiência do algoritmo em gerir este recurso. É interessante comentar que o tempo computacional, para a taxa de 48kbps, não apresentou valores consideravelmente maiores do que os obtidos pela técnica padrão do codificador de referência. Assim, identificou-se que as melhores soluções do Algoritmo Híbrido foram obtidas, com um tempo computacional considerado aceitável.

Os resultados do Algoritmo Genético também se mostraram melhores, em relação à qualidade perceptual obtida com a técnica *TLS*, para todos os arquivos de áudio teste, com exceção do número 4. Neste, a quantidade de distorção obtida com o AG foi superior à técnica *TLS*. Observa-se que o tamanho do arquivo 4 obtido com o AG, poderia ter sido aproximadamente 2500 bytes maior, comparando-se com o valor da Tabela 50. Isso demonstra que a taxa média de bits neste caso, foi levemente inferior à 48kbps. Nesta codificação, faltou ao processo de busca, encontrar soluções que utilizassem melhor o recurso da quantidade de bits. O tempo de processamento necessário para solucionar o problema de codificação de cada *frame* foi superior ao necessário pelo Algoritmo Híbrido, de forma considerável.

O algoritmo da BT obteve sucesso em melhorar a qualidade perceptual de sete dos dez arquivos de áudio teste. Observa-se que para os arquivos 4 e 8, seus respectivos tamanhos ficaram abaixo do valor que realmente poderiam estar, quando analisada a Tabela 50. Desta forma, com uma melhor utilização do recurso da quantidade de bits, a taxa média em kbps sofreria um aumento, de maneira que a qualidade perceptual poderia ser também aumentada. O tempo de processamento necessário pela BT foi maior que o tempo necessário pelo Algoritmo Híbrido e Genético. Isso se deve ao fato de que as boas soluções, foram encontradas com *BT_Max* igual a 500, o que prolonga o tempo da trajetória da busca, e conseqüentemente o tempo de processamento.

Para a taxa de 116kbps, os experimentos para os dez sons da biblioteca de áudio teste também foram realizados. Na Tabela 52 são apresentados os resultados obtidos com os quatro métodos de solução estudados para o Problema de

Codificação AAC. Os melhores resultados para cada arquivo de áudio teste foram destacados para facilitar sua identificação.

Método de Solução	Parâmetros de Avaliação	Índice	1	2	3	4	5	6	7	8	9	10
		Tipo de som	Bateria	Blues	Contra baixo	Coral	Crash China	Cravo	Narração	Orquestra	Rock	Sax
Codificador de Referência <i>TLS</i>	ANMR - FO		685	1.391	53	115	681	682	89	114	11.501	312
	Tamanho do arquivo		125.781	223.935	139.898	149.375	46.974	148.893	157.473	228.831	225.631	122.380
	Tempo de Processamento		35.963	34.039	34.226	34.902	33.064	35.003	35.852	34.886	34.978	36.054
Busca Tabu	ANMR - FO	μ	421	530	3	100	743	355	14	62	4.634	105
		σ	16	13	0,112	1	38	65	1	0,291	36	17
	Tamanho do arquivo	μ	123.291	223.667	140.854	150.376	46.772	147.353	158.046	227.783	222.375	122.196
		σ	822	233	192	230	90	1.838	240	387	301	584
	Tempo de Processamento	μ	404.206	359.114	435.141	436.184	434.351	440.000	448.801	465.084	403.217	447.554
		σ	3.375	8.912	9.514	13.562	10.459	20.818	9.132	10.313	9.367	50.263
Algoritmo Genético	ANMR - FO	μ	498	533	19	118	704	361	26	64	3.749	109
		σ	17	19	9	27	29	19	5	11	55	19
	Tamanho do arquivo	μ	121.557	212.775	130.182	142.562	44.693	142.873	147.861	217.663	215.226	116.258
		σ	1.296	1.428	1.315	1.184	802	984	1.946	1.519	390	1.496
	Tempo de Processamento	μ	135.663	160.133	140.937	151.105	149.848	165.140	150.075	159.735	159.227	167.880
		σ	1.497	1.530	2.505	5.877	7.367	2.970	3.712	5.048	3.493	4.296
Algoritmo Híbrido	ANMR - FO	μ	404	422	2	65	563	266	13	44	3.192	105
		σ	49	10	0,102	2	19	24	1	7	29	13
	Tamanho do arquivo	μ	121.050	222.301	140.591	148.674	46.809	147.163	157.578	227.605	220.177	120.157
		σ	844	526	306	203	154	434	286	323	428	280
	Tempo de Processamento	μ	133.443	132.446	162.645	112.884	157.642	126.674	191.689	132.376	208.428	207.668
		σ	2.699	3.131	9.356	3.293	13.818	3.658	7.564	15.957	3.343	14.190

Tabela 52 – Resultados da codificação para a taxa de 116kbps

Pode-se identificar que o Algoritmo Híbrido obteve novamente os menores valores de FO, para todos os dez arquivos de áudio teste. Os valores baixos de desvio padrão das soluções, demonstram a convergência e eficiência do algoritmo. Em relação ao tempo de processamento, observa-se um incremento significativo para o Algoritmo Híbrido, em relação ao tempo necessário para a codificação na taxa de 48kbps. Fato este, que não é observado com o tempo de processamento da técnica *TLS* do codificador de referência, que apresenta um aumento muito pequeno, da taxa de 48kbps para 116kbps.

O Algoritmo Genético apenas não obteve melhores resultados em relação à técnica *TLS* em dois dos dez arquivos de áudio teste, o de número 4 e 5. Observa-se que nestes dois arquivos, seus tamanhos ficaram aproximadamente 6000 e 2500

bytes respectivamente, abaixo do tamanho que poderiam obter. Este fato mostra que a taxa média de bits, utilizada nestes dois arquivos, ficou abaixo da taxa pré-definida para o experimento, de 116kbps. Este comportamento se deve ao AG não ter obtido sucesso, para uma parcela significativa dos *frames* do arquivo, em definir com mais eficiência as regiões promissoras do espaço de soluções. Desta forma, a busca acabou sendo restringida às soluções que não utilizaram por completo o recurso disponível de quantidade de bits, produzindo resultados com qualidade perceptual menor. O tempo de processamento necessário para a execução do Algoritmo Genético, para a taxa de 116kbps foi semelhante ao tempo do Algoritmo Híbrido, aproximadamente 100ms superior ao tempo do método *TLS*.

Para a taxa de 116kbps, a Busca Tabu obteve melhores resultados do que a técnica do codificador de referência, com exceção do arquivo de número 5. Este arquivo possui um som de um prato *china* de bateria, e desta forma, possui a característica de conter componentes espectrais de alta frequência. Sinais de maior frequência são complicados de serem codificados, e assim apresentam uma dificuldade maior para o codificador. Deve-se observar que o AG também não obteve sucesso para este arquivo, em melhorar o valor de FO encontrado pelo codificador de referência. O tempo de processamento necessário neste experimento foi consideravelmente superior aos demais métodos, com um incremento médio de aproximadamente 300ms em relação ao tempo obtido com o Algoritmo Híbrido.

Os experimentos seguintes foram realizados utilizando a taxa de 184kbps. Os resultados obtidos pelos quatro diferentes métodos de solução do problema estudado, *TLS*, Busca Tabu, Algoritmo Genético e Algoritmo Híbrido, são apresentados na Tabela 53. As soluções de melhor qualidade foram destacadas a fim de facilitar sua identificação.

Pode-se observar através da Tabela 53, que os melhores resultados para os dez sons da biblioteca de áudio teste foram encontrados pelo Algoritmo Híbrido. Novamente os valores de desvio padrão encontrados foram baixos, apresentando boa convergência nas soluções. Para obter as melhores soluções, o Algoritmo Híbrido precisou dispensar um tempo de processamento maior do que o utilizado pela técnica *TLS* e Algoritmo Genético.

Método de Solução	Parâmetros de Avaliação	Índice	1	2	3	4	5	6	7	8	9	10
		Tipo de som	Bateria	Blues	Contra baixo	Coral	Crash China	Cravo	Narração	Orquestra	Rock	Sax
Codificador de Referência TLS	ANMR - FO		116	171	3	24	191	81	4	23	1.084	19
	Tamanho do arquivo		195.781	346.403	205.514	230.705	73.173	230.648	237.863	354.082	346.909	190.390
	Tempo de Processamento		36.853	36.062	35.907	37.014	36.426	36.338	37.308	37.971	36.910	37.730
Busca Tabu	ANMR - FO	μ	122	132	1	15	133	103	4	11	748	16
		σ	3	3	0,025	0,195	30	2	0,089	0,110	9	1
	Tamanho do arquivo	μ	194.851	343.989	210.455	232.384	73.455	230.339	239.889	352.475	341.216	190.037
		σ	239	383	1.120	282	162	422	170	735	802	265
	Tempo de Processamento	μ	489.723	435.450	477.014	478.982	469.698	490.134	428.098	442.785	467.066	414.467
		σ	10.515	11.276	11.925	14.471	20.133	19.146	10.129	11.429	11.484	18.183
Algoritmo Genético	ANMR - FO	μ	169	136	20	35	94	79	11	24	702	24
		σ	46	32	11	12	9	15	2	9	63	5
	Tamanho do arquivo	μ	190.630	335.993	201.900	222.053	69.714	225.467	242.726	342.511	343.831	183.414
		σ	2.853	1.942	4.392	3.066	1.765	1.951	3.874	2.971	1.039	2.354
	Tempo de Processamento	μ	185.328	195.343	158.819	185.554	181.022	201.354	173.670	181.655	197.660	190.320
		σ	4.440	3.371	4.029	3.655	7.409	3.598	2.178	4.016	4.309	3.995
Algoritmo Híbrido	ANMR - FO	μ	94	82	0,416	10	92	52	3	8	551	11
		σ	7	3	0,041	0,233	8	2	0,487	0,226	7	1
	Tamanho do arquivo	μ	191.506	347.555	221.819	232.884	73.071	231.122	245.983	357.136	346.683	190.029
		σ	427	312	389	359	210	219	334	431	541	213
	Tempo de Processamento	μ	242.354	345.773	316.221	318.410	272.691	310.516	342.276	355.826	272.269	310.922
		σ	15.237	9.277	38.034	18.982	30.006	9.627	35.676	13.550	39.495	16.643

Tabela 53 – Resultados da codificação para a taxa de 184kbps

Para a taxa de 184kbps, o Algoritmo Genético conseguiu melhorar a solução do método *TLS* em quatro dos dez arquivos de áudio teste. Verifica-se que os tamanhos de arquivo estiveram abaixo do tamanho determinado para a taxa de 184kbps, em nove dos dez sons. Isso demonstra que a busca não conseguiu encontrar combinações entre as três variáveis de decisão, que utilizassem de forma eficiente o recurso disponível do processo de codificação, a quantidade de bits.

A Busca Tabu não obteve sucesso em três dos dez arquivos de áudio teste, em melhorar a solução encontrada pelo método *TLS* sugerido na norma. Porém, o tempo de processamento necessário em sua execução foi o maior dos métodos estudados. Isso se deve à configuração *BT_max* igual a 500, que prolongou a busca para produzir melhores resultados, e ao mesmo tempo incrementou o tempo computacional necessário para sua execução.

Por fim, os experimentos foram realizados para a taxa de 250kbps. Os resultados obtidos com os quatro métodos de solução abordados neste trabalho,

podem ser verificados na Tabela 54. Os melhores resultados foram destacados para uma fácil identificação.

Método de Solução	Parâmetros de Avaliação	Índice	1	2	3	4	5	6	7	8	9	10
		Tipo de som	Bateria	Blues	Contra baixo	Coral	Crash China	Cravo	Narração	Orquestra	Rock	Sax
Codificador de Referência <i>TLS</i>	ANMR - FO		21	48	2	6	39	18	3	4	354	4
	Tamanho do arquivo		263.111	477.882	306.651	312.660	98.013	312.898	318.852	490.374	469.397	255.548
	Tempo de Processamento		37.091	38.208	37.498	38.902	37.796	38.745	37.860	38.169	37.299	37.354
Busca Tabu	ANMR - FO	μ	24	27	0,127	3	23	17	1	2	166	3
		σ	1	1	0,004	0,123	1	0,354	0,041	0,068	2	0,113
	Tamanho do arquivo	μ	261.595	478.916	306.731	312.757	100.125	312.576	337.130	491.527	469.762	257.040
		σ	184	376	229	224	176	687	219	361	626	258
	Tempo de Processamento	μ	591.675	501.689	565.110	550.740	565.472	591.656	508.573	545.667	597.300	559.667
		σ	13.523	8.375	12.754	15.211	19.558	17.065	7.975	9.103	28.499	21.500
Algoritmo Genético	ANMR - FO	μ	45	59	6	9	32	30	7	18	218	15
		σ	19	24	3	2	8	5	2	5	59	5
	Tamanho do arquivo	μ	260.839	463.355	302.199	299.272	98.016	304.423	307.484	480.379	465.258	245.997
		σ	6.953	3.303	6.620	6.450	3.033	3.353	5.925	9.635	2.799	5.807
	Tempo de Processamento	μ	205.813	222.603	176.552	212.681	202.468	231.041	190.064	208.942	231.385	200.697
		σ	2.016	2.608	4.579	3.323	7.063	3.525	2.096	3.615	3.554	3.651
Algoritmo Híbrido	ANMR - FO	μ	14	16	0,106	2	15	10	0,593	2	102	2
		σ	1	0,497	0,002	0,059	1	0,260	0,027	0,034	2	0,189
	Tamanho do arquivo	μ	260.825	476.675	306.673	318.569	100.878	312.246	338.336	491.598	472.598	257.900
		σ	302	491	326	292	223	300	273	450	696	340
	Tempo de Processamento	μ	451.630	432.583	423.438	432.072	425.121	442.265	412.731	421.976	333.989	439.041
		σ	18.801	147.141	15.849	21.931	38.458	13.724	28.463	18.647	13.290	20.466

Tabela 54 – Resultados da codificação para a taxa de 250kbps

Novamente, verifica-se que os melhores resultados obtidos foram encontrados pelo Algoritmo Híbrido, para todos os dez sons dos arquivos de áudio teste. Os baixos valores de desvio padrão, também foram identificados nos resultados com a taxa de 250kbps, o que permite afirmar que Algoritmo Híbrido mostrou considerável precisão e convergência das soluções, para todas as quatro taxas de bits estudadas. O tempo de processamento necessário para a codificação em AAC na taxa 250kbps, utilizando o Algoritmo Híbrido, apresentou um aumento significativo, quando comparado ao método *TLS* do codificador de referência. Isso se deve à complexidade do problema ser maior quando a taxa de bits é aumentada. Desta forma, o espaço de soluções cresce, e assim as metaheurísticas permanecem mais tempo alternando entre as estratégias de intensificação e diversificação, aumentando o tempo de processamento necessário para solucionar o Problema de Codificação AAC.

Algoritmo Genético melhorou as soluções obtidas com a técnica *TLS* em dois dos dez arquivos de áudio teste. Neste caso, pode-se observar novamente que os tamanhos de arquivos gerados poderiam ter sido maiores. Desta forma, mais bits seriam distribuídos entre os frames, o que poderia ocasionar em um aumento da qualidade perceptual do áudio. Os tempos de processamento necessários para realizar as codificações foram menores que os obtidos com o Algoritmo Híbrido, porém ainda foram superiores aos valores da técnica do codificador de referência.

Em relação à Busca Tabu, pode-se observar que foram encontradas soluções de melhor qualidade do que da técnica do codificador de referência, em nove dos dez arquivos de áudio teste. Porém, deve-se observar que esta melhora, impactou com um aumento no tempo computacional necessário para a codificação de cada *frame*. A busca prolongada com *BT_Max* igual a 500 garante bons resultados, mas acaba determinando um considerável aumento do tempo total de processamento.

Com intuito de realizar uma avaliação geral, é apresentada na Tabela 55 e na Tabela 56, as relações em dB e percentuais respectivamente, dos métodos Busca Tabu, Algoritmo Genético e Algoritmo Híbrido, em relação à técnica *TLS* do codificador de referência MPEG-4 AAC. Pode-se identificar que a maioria dos campos da Tabela 55 é representada por uma relação negativa, que representa a diminuição dos valores obtidos da métrica *ANMR* (FO) através das metaheurísticas, em relação à técnica *TLS*. Em contrapartida, os valores de relação positiva identificam os casos em que as metaheurísticas não obtiveram sucesso em melhorar as soluções encontradas pela técnica do codificador de referência.

É importante destacar que as melhores relações encontradas, para os dez arquivos de áudio teste codificados, nas quatro taxas de bits estudadas, foram todas obtidas através do Algoritmo Híbrido. Desta forma, o Algoritmo Híbrido obteve melhores resultados em relação às soluções da *TLS*, em 100% dos 40 cenários estudados. Além de obter melhores resultados em relação à técnica do codificador de referência, os valores de desvio padrão identificados nas Tabelas 51, 52, 53 e 54, dão confiabilidade e precisão de que bons resultados para *ANMR* serão encontrados. Em relação à Busca Tabu, o algoritmo obteve melhores resultados que a *TLS*, em 80% dos cenários estudados, enquanto que o Algoritmo Genético melhorou os valores obtidos através da *TLS* em 57% dos cenários.

Medida de ANMR dos Algoritmos em relação à técnica TLS (dB)											
Taxa de Bits	Algoritmos	1	2	3	4	5	6	7	8	9	10
		Bateria	Blues	Contra baixo	Coral	Crash China	Cravo	Narração	Orquestra	Rock	Sax
48kbps	Busca Tabu	-1,31	-6,04	-5,21	1,63	-1,29	-3,36	-10,53	0,44	-4,04	0,19
	Algoritmo Genético	-2,51	-7,43	-4,18	0,56	-2,52	-4,54	-8,51	-0,40	-5,19	-0,37
	Algoritmo Híbrido	-2,89	-8,05	-5,68	-0,99	-3,53	-5,55	-10,89	-2,02	-6,15	-1,87
116kbps	Busca Tabu	-4,23	-8,38	-24,94	-1,21	0,76	-5,67	-16,07	-5,29	-7,90	-9,46
	Algoritmo Genético	-2,77	-8,33	-8,91	0,22	0,29	-5,53	-10,69	-5,01	-9,74	-9,13
	Algoritmo Híbrido	-4,59	-10,36	-28,46	-4,96	-1,65	-8,18	-16,71	-8,27	-11,13	-9,46
184kbps	Busca Tabu	0,44	-2,25	-9,54	-4,08	-3,14	2,09	0,00	-6,41	-3,22	-1,49
	Algoritmo Genético	3,27	-1,99	16,48	3,28	-6,16	-0,22	8,79	0,37	-3,77	2,03
	Algoritmo Híbrido	-1,83	-6,38	-17,16	-7,60	-6,34	-3,85	-2,50	-9,17	-5,88	-4,75
250kbps	Busca Tabu	1,16	-5,00	-22,01	-6,02	-4,59	-0,50	-9,54	-6,02	-6,58	-2,50
	Algoritmo Genético	6,62	1,79	11,48	3,52	-1,72	4,44	7,36	13,06	-4,21	11,48
	Algoritmo Híbrido	-3,52	-9,54	-23,58	-9,12	-8,30	-5,11	-14,08	-6,02	-10,81	-6,02

Tabela 55 – Comparação em dB entre as metaheurísticas e a técnica TLS para o Problema de Codificação AAC

Medida de ANMR dos Algoritmos em relação à técnica TLS (%)											
Taxa de Bits	Algoritmos	1	2	3	4	5	6	7	8	9	10
		Bateria	Blues	Contra baixo	Coral	Crash China	Cravo	Narração	Orquestra	Rock	Sax
48kbps	Busca Tabu	-13,99%	-50,13%	-45,09%	20,61%	-13,79%	-32,06%	-70,24%	5,25%	-37,23%	2,21%
	Algoritmo Genético	-25,09%	-57,50%	-38,18%	6,65%	-25,17%	-40,69%	-62,44%	-4,50%	-44,98%	-4,14%
	Algoritmo Híbrido	-28,32%	-60,40%	-48,00%	-10,73%	-33,41%	-47,22%	-71,46%	-20,77%	-50,74%	-19,34%
116kbps	Busca Tabu	-38,54%	-61,90%	-94,34%	-13,04%	9,10%	-47,95%	-84,27%	-45,61%	-59,71%	-66,35%
	Algoritmo Genético	-27,30%	-61,68%	-64,15%	2,61%	3,38%	-47,07%	-70,79%	-43,86%	-67,40%	-65,06%
	Algoritmo Híbrido	-41,02%	-69,66%	-96,23%	-43,48%	-17,33%	-61,00%	-85,39%	-61,40%	-72,25%	-66,35%
184kbps	Busca Tabu	5,17%	-22,81%	-66,67%	-37,50%	-30,37%	27,16%	0,00%	-52,17%	-31,00%	-15,79%
	Algoritmo Genético	45,69%	-20,47%	566,67%	45,83%	-50,79%	-2,47%	175,00%	4,35%	-35,24%	26,32%
	Algoritmo Híbrido	-18,97%	-52,05%	-86,13%	-58,33%	-51,83%	-35,80%	-25,00%	-65,22%	-49,17%	-42,11%
250kbps	Busca Tabu	14,29%	-43,75%	-92,06%	-50,00%	-41,03%	-5,56%	-66,67%	-50,00%	-53,11%	-25,00%
	Algoritmo Genético	114,29%	22,92%	275,00%	50,00%	-17,95%	66,67%	133,33%	350,00%	-38,42%	275,00%
	Algoritmo Híbrido	-33,33%	-66,67%	-93,38%	-65,00%	-61,54%	-44,44%	-80,23%	-50,00%	-71,19%	-50,00%

Tabela 56 – Comparação percentual entre as metaheurísticas e a técnica TLS para o Problema de Codificação AAC

8 CONCLUSÃO

Neste trabalho foi abordado o Problema de Codificação AAC, que utiliza como métrica na Função Objetivo a qualidade perceptual *ANMR*, e como principal restrição a taxa de bits na saída do codificador. Para abordar tal problema, foi desenvolvido um modelo de otimização baseado em uma estrutura híbrida formada pelas metaheurísticas Algoritmo Genético e Busca Tabu, para o codificador MPEG-4 AAC. Atualmente, este problema é abordado utilizando-se a técnica do codificador de referência MPEG-4 AAC, conhecida como *Two Loop Search (TLS)*. Porém, este método não resolve o problema de codificação de maneira ótima, e com garantia de convergência das soluções.

O modelo computacional utilizado na implementação da estrutura de otimização do Problema de Codificação AAC foi validado através de experimentos específicos, para avaliar a relação das variáveis de decisão com a métrica *ANMR* e com a taxa de bits. Os resultados obtidos apresentaram comportamento conforme a dinâmica do codificador MPEG-4 AAC, expressa na sua documentação de normatização (ISO/IEC, 2004) (ISO/IEC, 2005), e assim o modelo computacional pôde ser validado.

Para realização dos experimentos, foi elaborada uma biblioteca de arquivos de áudio teste, contemplando diferentes tipos de sons e estilos de música. O modelo de otimização foi desenvolvido em linguagem C ANSI, sendo inserido no código do codificador de referência desenvolvido pelo projeto de software livre FFmpeg, e foi executado em um processador Intel Core i3, sobre o sistema operacional Linux Ubuntu.

Através de uma análise dos experimentos, verifica-se que a parametrização correta das metaheurísticas pôde conduzir suas buscas para soluções de boa qualidade. Em relação à Busca Tabu, pôde-se observar que a diversificação da busca, obtida com o aumento do tamanho da Lista Tabu, promoveu as melhores soluções da metaheurística. Por outro lado, valores elevados de *BT_Max* prolongaram a busca, aumentando desta forma, o tempo de processamento, sem produzir benefícios consideráveis sobre as soluções. Outro ponto interessante que deve ser observado nos experimentos da Busca Tabu, operando separadamente, é que os melhores resultados foram encontrados com o tamanho da Lista Tabu igual a 200. Nos experimentos do Algoritmo Híbrido, os melhores valores de FO foram

obtidos com tamanho da Lista Tabu igual a 100. Esta diminuição do valor da Lista Tabu, na configuração do Algoritmo Híbrido em relação à Busca Tabu isolada, é justificada pelo fato de no Algoritmo Híbrido a maior diversificação da busca ter sido gerada pelo Algoritmo Genético. Assim, não houve necessidade de maiores valores para a Lista Tabu com o intuito de diversificação, pois dentro da estrutura híbrida de otimização o Algoritmo Genético acabou sendo responsável por esta tarefa.

Em relação à parametrização do Algoritmo Genético, observa-se que através de um aumento da taxa de mutação, a qualidade das soluções também foi incrementada. Isso ocorreu devido à restrição aplicada sobre os valores de *scalefactors* no processo de mutação. Identificou-se através de experimentos, que a busca é melhor conduzida pelo espaço de soluções, quando há pouca variação sobre os valores dos *scalefactors*, em muitas *scalefactors bands*. Este comportamento se deve ao fato dos *scalefactors* serem codificados diferencialmente, caracterizando uma interdependência de seus valores entre as *SFBs*. Observou-se também, que o Algoritmo Genético apresentou rápida convergência nas soluções, não havendo necessidade de seu parâmetro Número de Gerações, ser superior a 50 iterações. Isso demonstra, que rapidamente o processo de diversificação da busca, conseguiu identificar as áreas mais promissoras do espaço de soluções, e intensificá-la nestas regiões.

Analisando os resultados obtidos através do Algoritmo Híbrido, identifica-se que este obteve os melhores resultados para todos os dez sons dos arquivos de áudio teste, quando comparado à Busca Tabu, Algoritmo Genético e o algoritmo *TLS*, para as quatro taxas de bits estudadas. Ao unir a estratégia de intensificação da Busca Tabu com a estratégia de diversificação utilizada no Algoritmo Genético, a busca foi eficientemente conduzida pelo espaço de soluções, explorando novas regiões, e intensificando a procura em áreas promissoras. Com a união das duas metaheurísticas no Algoritmo Híbrido, obteve-se um novo algoritmo com um desempenho superior. Ao compará-lo com as metaheurísticas em separado, verifica-se que foi possível encontrar soluções melhores em menos tempo.

A convergência e precisão do Algoritmo Híbrido também devem ser destacadas, pois os baixos valores de desvio padrão verificados nas soluções, conferem confiabilidade e demonstram eficiência do algoritmo na obtenção de bons

resultados de qualidade perceptual para os arquivos codificados. Estas são qualidades importantes, pois apesar de estocásticas, as metaheurísticas na estrutura híbrida obtiveram resultados consideravelmente melhores do que a técnica *TLS* do codificador de referência, que é um método exato.

Em relação ao Algoritmo Híbrido, pode-se notar que com o aumento da taxa de bits, há também um aumento no tempo computacional necessário para resolver o Problema de Codificação AAC. Isso ocorre, pois quanto maior o valor da taxa de bits, maiores são as possibilidades de combinação entre as variáveis de decisão: *scalefactors*, Códigos de Huffman e agrupamentos dos blocos, caso o *frame* seja configurado como *short block*. A principal restrição do modelo computacional, é a taxa de bits, uma vez que este limite é aumentado, o espaço de soluções factíveis cresce, e desta maneira, para encontrar boas soluções, acaba sendo necessário um tempo computacional maior. Isso explica o tempo adicional necessário pelas metaheurísticas, quando comparadas ao método *TLS*. Quanto maior a taxa de bits, maior foi a diferença entre o tempo computacional obtido com o Algoritmo Híbrido e a técnica *TLS*. Ao mesmo tempo, maior foi o ganho em qualidade perceptual obtida pelo Algoritmo Híbrido, em relação ao método *Two Loop Search*.

Pode-se comentar também, que as duas metaheurísticas que compuseram a estrutura do Algoritmo Híbrido tiveram uma parametrização mais focada na estratégia de diversificação. A Busca Tabu através da Lista Tabu, e o Algoritmo Genético através do percentual de Mutação. Isso permite afirmar, através destes experimentos, que o Problema de Codificação AAC apresenta melhores resultados quando as estratégias de diversificação são exploradas com maior intensidade.

Em uma análise final, deve-se observar que a principal contribuição deste trabalho foi o desenvolvimento de um modelo de otimização integrado ao codificador MPEG-4 AAC, baseado em uma solução híbrida das metaheurísticas Algoritmo Genético e Busca Tabu. Assim, o modelo implementado, integrado ao codificador, gerou em sua saída arquivos de áudio comprimidos, no formato AAC, com uma qualidade perceptual superior à obtida pelo codificador AAC de referência. Devido ao tempo de processamento de cada *frame* ser superior ao tempo de um *frame* (43ms), o método implementado não é o mais adequado para aplicações de tempo real, como por exemplo, de radio e TV *broadcast*. Porém, para aplicações de

compressão *off-line*, como por exemplo, a realização de *backup* de CDs, ou a conversão do formato de arquivos *WAV*, para o formato AAC, com o objetivo de utilizá-los em dispositivos móveis de reprodução de música, o codificador com a estrutura do Algoritmo Híbrido é recomendado para ser utilizado. Desta forma, irá garantir e proporcionar uma qualidade perceptual de áudio superior, à obtida pelo codificador MPEG-4 AAC de referência.

Por fim, existem pontos que podem ser melhorados no trabalho desenvolvido, e outros que podem ser destacados a fim de ampliar o estudo da aplicação de metaheurísticas, na abordagem do Problema de Codificação AAC do codificador MPEG-4. Desta forma, como trabalhos futuros alguns comentários podem ser realizados:

1. Alteração das características das metaheurísticas. Em relação à Busca Tabu, podem ser estudadas diferentes estratégias de intensificação, através de alterações no processo de geração de vizinhança, a fim de melhorar o desempenho do algoritmo, focando principalmente na diminuição do tempo de processamento. Quanto ao Algoritmo Genético, outros processos de mutação poderiam ser estudados, como por exemplo, a aplicação de uma mesma variação aleatória sobre todas as *SFBs*, para que a interdependência entre as mesmas já pré-estabelecida, pudesse ser mantida. Outra possibilidade interessante seria um processo de mutação adaptativo, em função de uma métrica, que mede o quanto diversificada se encontra uma determinada população. Desta forma, a diversidade da população poderia ser mantida constante.

2. Algoritmo da *TLS* operando em conjunto com as metaheurísticas. O Algoritmo Híbrido poderia ser estudado, utilizando a técnica *TLS* do codificador de referência, participando ativamente no processo busca. Por ter uma característica de tempo de processamento baixo, seria interessante estudar o seu comportamento operando em conjunto com o Algoritmo Genético ou com a Busca Tabu. Assim, por exemplo, o Algoritmo Genético poderia ser responsável por identificar as regiões promissoras do espaço de soluções, para que a técnica *TLS* pudesse realizar sua busca local, e intensificar o processo de busca.

3. Consideração da quarta variável de decisão do modelo computacional. A variável de configuração de cada *frame*, em *long blocks* ou *short blocks*, poderia ser testada, como uma variável de decisão das metaheurísticas. Assim, seria possível identificar se os resultados obtidos com as metaheurísticas, gerenciando também esta variável, seriam melhores ou piores do que quando gerenciada pela métrica da Entropia Perceptual. Neste caso, o tempo de processamento é um parâmetro que também deve ser analisado, principalmente, pelas estruturas de soluções serem muito diferentes entre os *long blocks* e *short blocks*, o que provoca uma dificuldade maior de manipular os processos de geração de vizinhança e cruzamento de indivíduos.

4. Estratégias para diminuição do tempo de processamento das metaheurísticas. Seria interessante estudar maneiras e técnicas para redução do tempo de processamento do Algoritmo Híbrido desenvolvido, porém sem que estas alterações impactassem sobre a qualidade perceptual do sinal de áudio codificado.

REFERÊNCIAS BIBLIOGRÁFICAS

ABNT, A. B. D. N. T. **Televisão digital terrestre — Codificação de Vídeo, Áudio e Multiplexação. Parte 2: Codificação de Áudio**. Rio de Janeiro, p. 16. 2007.

AGGARWAL, A.; REGUNATHAN, S. L.; ROSE, K. **Trellis-Based Optimization of MPEG-4 Advanced Audio Coding**. IEEE Workshop on Speech Coding. Genoa: IEEE. 2000. p. 142-144.

AGGARWAL, A.; REGUNATHAN, S. L.; ROSE, K. **Near - Optimal Selection of Encoding Parameters for Audio Coding**. ICASSP: International Conference on Acoustics, Speech, and Signal Processing. Salt Lake: IEEE. 2001. p. 3269-3272.

ALI, M. T.; SALEEM, M. **Improved Audio Quality at 48 Kbits/s for MPEG-4 AAC**. Second International Conference on Electrical Engineering. Lahore: IEEE. 2008. p. 1-8.

BAUER, C. The Optimal Choice of Encoding Parameters for MPEG4 AAC Streamed over Wireless Networks. **ACM - Wireless Multimedia Networking and Performance Modeling**, Nova Iorque, p. 93-100, 13 Outubro 2005.

BAUER, C.; FELLERS, M.; DAVIDSON, G. **Multidimensional Optimization of MPEG-4 AAC Encoding**. ICASSP: International Conference on Acoustics, Speech, and signal Processing. Toulouse: IEEE. 2006. p. 69-72.

BAUER, C.; VINTON, M. Joint Optimization of Scale Factors and Huffman Code Books for MPEG-4 AAC. **IEEE Transactions On Signal Processing**, v. 54, n. 1, p. 177-189, Janeiro 2006.

BELLARD, F. FFmpeg Project. Disponível em: <<http://www.ffmpeg.org/download.html>>. Acesso em: 15 fevereiro 2013.

BOSI, M.; GOLDBERG, R. E. **Introduction to Digital Audio Coding and Standards**. 1ª. ed. Massachusetts: Kluwer Academic Publishers, 2003.

BRANDENBURG, K. **MP3 and AAC Explained**. 17^a International Conference on High Quality Audio Coding. Florence: AES (Audio Engineering Society). 1999. p. 1-12.

CARTINHOOR, J. **Digital Signal Processing**. Nova Jersey: Prentice Hall, 2000.

COVER, T. M.; THOMAS, J. A. **Elements Of Information Theory**. 2^a. ed. New Jersey: John Wiley & Sons, 2006.

DERRIEN, O.; CHARBIT, M.; DUHAMEL, P. **Statistical Model for the Quantization Noise in the MPEG Advanced Audio Coder**. Proceeding of the IEEE International Conference on Acoustics Speech and Signal Processing. Orlando: IEEE. 2002. p. 1849-1852.

DERRIEN, O.; DUHAMEL, P.; CHARBIT, M. A New Quantization Optimization Algorithm for the MPEG Advanced Audio Coder Using a Statistical Subband Model of the Quantization Noise. **IEEE Transactions on Audio, Speech, and Language Processing**, Paris, v. 14, n. 4, p. 1328-1339, Julho 2006.

DUTOIT, T.; MARQUÉS, F. **Applied Signal Processing: A MATLAB - Based Proof of Concept**. New York: Springer, 2009.

FASTL, H.; ZWICKER, E. **Psychoacoustics: Facts and Models**. 3^a. ed. New York: Springer, 2007.

FISCHER, W. **Digital Video and Audio Broadcasting Technology: A Practical Engineering Guide**. 3^a. ed. Nova Iorque: Springer, 2010.

GAREY, M. R.; JOHNSON, D. S.; SETHI, R. The Complexity of Flowshop and Jobshop Scheduling. **Mathematics of Operations Research**, v. 1, n. 2, p. 117-129, Maio 1976.

GENDREAU, M.; HERTZ, A.; LAPORTE, G. A Tabu Search Heuristic for the Vehicle Routing Problem. **Management Science**, v. 40, n. 10, p. 1276-1290, Outubro 1994.

GENDREAU, M.; LAPORTE, G.; POTVIN, J.-Y. Metaheuristics for the capacitated VRP. **The vehicle routing problem**, Filadélfia, v. 9, p. 129-154, 2001.

GLOVER, F. Future Paths for Integer Programming And Links to Artificial Intelligence. **Computers & Operations Research**, v. 13, n. 5, p. 533-549, 1986.

GLOVER, F. Tabu Search - Part I. **Operations Research Society of America: Journal on Computing**, v. 1, n. 3, p. 17, 1989.

GLOVER, F. Tabu Search: Part II. **Operations Research Society of America: Journal on Computing**, v. 2, n. 1, p. 29, 1990.

GLOVER, F.; KELLY, J. P.; LAGUNA, M. Genetic Algorithms and Tabu Search: Hybrids for Optimization. **Computers & Operations Research**, v. 22, n. 1, p. 111-134, 1995.

GLOVER, F.; LAGUNA, M. **Tabu Search**. Norwell: Kluwer Academic Publishers, 1997.

GOLDBERG, D. E. **Genetic algorithms in search, optimization, and machine learning**. University of Michigan: Addison-Wesley Pub. Co. 1989.

GÓMEZ, A. T. et al. **Especificação Técnica e Sistêmica - Plataforma de Convergência Digital IPTV/TV Digital**. Universidade do Vale do Rio dos Sinos - Unisinos. São Leopoldo, p. 74. 2011. (ISSN 2332/09).

GÓMEZ, A. T.; LINCK, I. C. C. **A Hybrid Algorithm using Metaheuristics Applied to H.264/AVC Video Encoder**. 9th International Conference on Informatics in Control, Automation and Robotics. Roma: SciTePress. 2012. p. 212-217.

GRIMMETT, G.; STIRZAKER, D. **Probability and Random Processes**. 3^a. ed. Nova Iorque: Oxford University Press, 2001.

HANZO, L.; SOMERVILLE, F. C.; WOODARD, J. **Voice and Audio Compression for Wireless Communications**. 2^a. ed. Chippenham: John Wiley & Sons, 2007.

HAYES, M. H. **Processamento Digital de Sinais**. Porto Alegre: Bookman, 2006.

HENRIQUE, L. L. **Acústica Musical**. Lisboa: Fundação Calouste Gulbenkian, 2002.

HERRE, J. **Temporal Noise Shaping, Quantization and Coding Methods in Perceptual Audio Coding: a Tutorial Introduction**. 17^a International Conference on High Quality Audio Coding. Florence: AES. 1999. p. 14.

HOFFMANN, G. A. **Study of the Audio Coding Algorithm of the MPEG-4 AAC Standard and Comparison Among Implementations of Modules of the Algorithm**. Dissertação (Mestrado em Ciências da Computação) - Universidade Federal do Rio Grande do Sul. Porto Alegre, p. 115. 2002.

HOLLAND, J. H. **Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence**. 1. ed. Michigan: University of Michigan Press, 1975.

ISO/IEC. **11172-3: Part 3 Audio - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 MBIT/s**. ISO. Geneva. 1993.

ISO/IEC. **13818-3: Information technology - Generic Coding of Moving Pictures and Associated Audio Information - Part 3: Audio**. ISO. Genebra. 1998.

ISO/IEC. **13818-7: Information technology — Generic Coding of Moving Pictures and Associated Audio Information : Part 7 - Advanced Audio Coding (AAC)**. ISO. Genebra, p. 206. 2004.

ISO/IEC. **14496-3: Information Technology - Coding of Audio Visual Objects - Part 3: Audio**. ISO. Genebra, p. 1172. 2005.

JADAAN, O. A.; RAJAMANI, L.; RAO, C. R. Improved Selection Operator for GA. **Journal of Theoretical and Applied Information Technology**, Osmania, 2008. 269-277.

JEON, G.; LEEP, H. R. Forming Part Families by Using Genetic Algorithm and Designing Machine Cells Under Demand Changes. **Computers & Operations Research**, n. 33, p. 263-283, Abril 2005.

JOHNSTON, J. D. Transform Coding of Audio Signals Using Perceptual Noise Criteria. **IEEE Journal on Selected Areas in Communications**, v. 6, n. 2, p. 314-323, Fevereiro 1988.

KANDADAI, S. A. **Perceptual Audio Coding That Scales to Low Bitrates**. Dissertação (Doutorado em Filosofia) - Universidade do Novo México. Las Cruces, p. 193. 2007.

KUSIAK, A.; DORF, A.; RICHARD, C. **Handbook of Design, Manufacturing and Automation**. New York: John Wiley and Sons, 1994.

LAN-YING, L.; MIN, S. Software-Hardware Partitioning Strategy Using Hybrid Genetic and Tabu Search. **IEEE - International Conference on Computer Science and Software Engineering**, p. 83-86, 2008.

LAW, A. M.; KELTON, W. D. **Simulation Modeling and Analysis**. 3ª. ed. Arizona: Mcgraw-Hill, 2000.

LI, T.; RAHARDJA, S. Transparent Bitrate Estimation for Perceptual Audio Coding. **ICIEA 2009 - IEEE Conference on Industrial Electronics and Applications**, Xi'an, Julho 2009. 2123-2127.

MELTZER, S.; MOSER, G. MPEG 4 HE-AAC v2: Audio Coding for Today's Digital Media World. **EBU Technical Review**, p. 1-12, Janeiro 2006.

MOCK, T. It's All About the Algorithm-But Which One Will Win? **IEEE Spectrum: Technology & Science News**, p. 42-47, Setembro 2004.

MORIYA, T.; IWAKAMI, N. A Design of Lossy and Lossless Scalable Audio Coding. **IEEE**, p. 889-892, 2000.

MURATA, T.; ISHIBUCHI, H. **Performance Evaluation of Genetic Algorithms for Flowshop Scheduling Problems**. World Congress on Computational Intelligence. Orlando: IEEE. 1994. p. 812-817.

NIAMUT, O. A. **Rate-Distortion Optimal Time-Frequency Decompositions for MDCT-based Audio Coding**. Tese (Doutorado em Engenharia Elétrica) - Universidade Técnica de Delft. Delft, p. 196. 2006.

OPPENHEIM, A. V.; SCHAFER, R. W. **Discrete-time Signal Processing**. Nova Jersey: Prentice Hall, 1989.

PAINTER, T.; SPAN, A. Perceptual Coding of Digital Audio. **Proceedings of the IEEE**, Raleigh, v. 88, n. 4, p. 451-513, Abril 2000.

PEREIRA, F.; EBRAHIMI, T. **The MPEG-4 Book**. Nova Jersey: Prentice Hall, 2002.

POHLMAN, K. C. **Principles of Digital Audio**. New York: McGraw-Hill, 1995.

POLTOSI, M. R. **Elaboração de Escalas de Trabalho de Técnicos de Enfermagem com Busca Tabu e Algoritmo Genético**. Tese (Mestrado em Computação Aplicada) – Universidade do Vale do Rio dos Sinos - Unisinos. São Leopoldo. 2007.

RAVELLI, E. et al. Joint Optimization of the Perceptual Core and Lossless Compression Layers in Scalable Audio Coding. **ICASSP 2010 - International Conference on Acoustics, Speech, and Signal Processing**, Dallas, Março 2010. 365-368.

REEVES, C. R. Improving the Efficiency of Tabu Search for Machine Sequencing Problem. **Operational Research Society**, v. 44, n. 4, p. 375-382, 1993.

SANDVINE. **Global Internet Phenomena Report: Fall 2011**. Sandvine: Intelligent Broadband Networks. Waterloo, p. 1-23. 2011.

SANDVINE. **The Global Internet Phenomena Report: 2H 2012**. Sandvine: Intelligent Broadband Networks. Waterloo, p. 1-35. 2012.

SINGH, N. C. Measuring the 'Complexity' of Sound. **Pramana - Journal of Physics**, Manesar, v. 77, n. 5, p. 811-816, Novembro 2011.

SRIVASTAVA, B.; CHEN, W.-H. Part Type Selection Problem in Flexible Manufacturing Systems: Tabu Search Algorithms. **Annals of Operations Research**, v. 41, n. 3, p. 279-297, 1993.

STOLFI, G. **Princípios de Televisão Digital**. Universidade Presbiteriana Mackenzie. São Paulo, p. 1-24. 2008.

TANKANOW, A. S. **The Automatic Extraction of a Measure of Complexity from an Audio Sound File**. Tese (Mestrado em Engenharia da Música) - Universidade de Miami. Miami, p. 116. 2007.

VIANA, V. **Metaheurísticas e Programação Paralela em Otimização Combinatória**. Fortaleza: Edições UFC, 1998.

WANG, B.; ZHANG, J.; YAO, Y. **A New Bit-allocation Algorithm for AAC Encoder Based on Linear Prediction**. 11^a International Conference on Communication Technology Proceedings. Hangzhou: IEEE. 2008. p. 726-729.

WATKINSON, J. **The Art of Digital Audio**. 3^a. ed. Oxford: Focal Press, 2001.

WICKERT, T. I.; GÓMEZ, A. T. **An Architecture based on Ontologies, Agents and Metaheuristics Applied to the Multimedia Service of the Brazilian Digital Television System**. DATA 2012 - International Conference on Data Technologies and Applications. Roma: SciTePress. 2012. p. 203-208.

WU, S.; QIU, X. **An Bit Allocation Method Based Rate-Distortion Control Algorithm for MPEG-4 Advanced Audio Coding**. ICALIP2008 - International Conference on Audio, Language and Image Processing. Shanghai: IEEE. 2008. p. 237-241.

YANG, C.-H.; HANG, H.-M. **Cascaded Trellis-Based Rate-Distortion Control Algorithm for MPEG-4 Advanced Audio Coding**. **IEEE Transactions on Audio, Speech, and Language Processing**, v. 14, n. 3, p. 998-1007, Maio 2006.