



Programa Interdisciplinar de Pós-Graduação em

# **Computação Aplicada**

**Mestrado Acadêmico**

Iris Correa das Chagas Linck

Um Sistema de Codificação de Vídeo para TV Digital -  
SBTVD

São Leopoldo, 2012

UNIVERSIDADE DO VALE DO RIO DOS SINOS  
CIÊNCIAS EXATAS E TECNOLÓGICAS  
PROGRAMA INTERDISCIPLINAR DE PÓS-GRADUAÇÃO  
EM COMPUTAÇÃO APLICADA – PIPCA  
NÍVEL MESTRADO

IRIS CORRÊA DAS CHAGAS LINCK

**UM SISTEMA DE CODIFICAÇÃO DE VÍDEO PARA TV DIGITAL - SBTVD**

São Leopoldo

2012

IRIS CORRÊA DAS CHAGAS LINCK

**UM SISTEMA DE CODIFICAÇÃO DE VÍDEO PARA TV DIGITAL – SBTVD**

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

Área de atuação: Pesquisa Operacional

Orientador: Prof. Dr. Arthur Tórgo Gómez

São Leopoldo

2012



Ficha catalográfica

L736s Linck, Iris Correa das Chagas  
Um sistema de codificação de vídeo para TV digital –  
SBTVD / por Iris Correa das Chagas Linck. – 2012.  
106 f. : il., 30 cm.

Dissertação (mestrado) — Universidade do Vale do Rio  
dos Sinos, Programa de Pós-Graduação em Computação  
Aplicada, 2012.  
Orientação: Prof. Dr. Arthur Tórgo Gómez.

1. Metaheurística. 2. Busca tabu. 3. Algoritmo genético.  
4. Codec H264. I. Título.

CDU 004:519.8

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

IRIS CORRÊA DAS CHAGAS LINCK

**UM SISTEMA DE CODIFICAÇÃO DE VÍDEO PARA TV DIGITAL – SBTVD**

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

Área de atuação: Ciências Exatas e da Terra

Orientador: Prof. Dr. Arthur Tórgo Gómez

Aprovado em \_\_\_ de \_\_\_\_\_ de 2012.

BANCA EXAMINADORA

---

Prof. Dr. Arthur Tórgo Gómez

PIPCA – Universidade do Vale do Rio dos Sinos – UNISINOS

---

Prof. Dr. José Vicente Canto dos Santos

PIPCA – Universidade do Vale do Rio dos Sinos - UNISINOS

---

Prof. Dr. Plácido Rogério Pinheiro

Universidade de Fortaleza - UNIFOR

## **AGRADECIMENTOS**

Agradeço em especial ao meu orientador, Dr. Arthur Tórgo Gómez, uma pessoa admirável que me mostrou o mundo da pesquisa e acreditou no meu potencial.

Ao professor Dr. Leonardo Dagnino Chiwiacowsky que várias vezes me auxiliou tirando dúvidas sobre Algoritmo Genético.

À FINEP pelo apoio financeiro.

Aos Professores e ao laboratorista do PIPCA que foram sempre atenciosos.

Aos colaboradores do projeto DigConv da UNISINOS.

Aos chefes e colegas de trabalho pelo apoio e folgas concedidas.

Ao meu marido pelo apoio e compreensão.

## RESUMO

Neste trabalho é desenvolvido um algoritmo híbrido que simula o comportamento do Codificador/Decodificador de vídeo H.264/AVC, ou simplesmente CODEC H.264, utilizado no Sistema Brasileiro de Televisão Digital. O algoritmo proposto tem a finalidade de buscar a melhor configuração possível de seis dos principais parâmetros utilizados para a configuração do CODEC H.264. Este problema é abordado como um problema de otimização combinatória conhecido como Problema de Seleção de Partes e que é classificado como NP-Difícil. O algoritmo híbrido proposto, denominado Simulador de Metaheurísticas aplicado a um CODEC (SMC), foi desenvolvido com base em duas metaheurísticas: Busca Tabu e Algoritmo Genético. Os seis parâmetros de configuração a serem otimizados pelo SMC são: o *bit rate*; o *frame rate*; os parâmetros de quantização de quadros tipo B, tipo P e tipo I e a quantidade de quadros tipo B em um grupo de imagens (GOP – Group of Pictures). Os dois primeiros parâmetros mencionados atuam basicamente sobre a qualidade da imagem do vídeo enquanto que os demais parâmetros atuam diretamente na compressão do vídeo. Experimentos e testes foram feitos utilizando-se o CODEC H.264 desenvolvido no Projeto Plataforma de Convergência Digital IPTV/TV Digital (DigConv). Nos experimentos o CODEC tem seus parâmetros configurados de acordo com os resultados obtidos pelo SMC. Um vídeo é codificado no CODEC H.264 para que se possa analisar a sua qualidade de imagem e o seu grau de compressão após o processo de codificação. É feita uma correlação entre esses resultados e a Função Objetivo do SMC. A qualidade da imagem é medida através da métrica mais utilizada na literatura, o PSNR (*Peak Signal to Noise Ratio*), que é calculada pelo próprio CODEC ao final da codificação de um vídeo. Verificou-se que à medida que a Função Objetivo aumenta, o CODEC H.264 consegue obter uma melhor qualidade de imagem e um maior grau de compressão de vídeo.

**Palavras chave:** Metaheurísticas. Busca Tabu. Algoritmo Genético. Codec H.264.



## ABSTRACT

In this work is developed a hybrid algorithm that simulates the behavior of the H.264/AVC video encoder/decoder, or simply H.264 video CODEC, used in the Brazilian System of Digital Television. The proposed algorithm intends to seek the best possible configuration of the six main parameters used for configuring the H.264 video CODEC. This problem is treated as a combinatorial optimization problem known as the Parties Selection Problem, which is classified as NP-Hard. The proposed hybrid algorithm, called Simulator Metaheuristics applied to a CODEC (SMC), was developed based on two metaheuristics: Tabu Search and Genetic Algorithm. The six configuration parameters to be optimized by the SMC are the *bit rate*, *frame rate*, the parameters of quantization tables of type B, type I and type P and the amount of frames type B in a group of pictures (GOP - Group of Pictures). The first two parameters mentioned, work primarily on the quality of the video image while the other parameters act directly on the video compression. Experiments and tests were done using the video CODEC H.264 developed in Digital Convergence Platform IPTV/Digital TV Project (DigConv). DigConv Project. In the experiments the CODEC has its parameters set according to the results obtained by the SMC. Then, a video is encoded by the CODEC in order to analyze the video image quality and the video compression degree reached after the encoding process. It is made a correlation between these results and the objective function of the SMC. The picture quality is measured by the metric most often used in literature, the PSNR (Peak Signal to Noise Ratio), which is calculated by the CODEC at the end of a video encoding process. It was found that as the objective function has increased, the CODEC reached a better image quality and a higher video compression.

**Keywords:** Metaheuristics, Tabu Search, Genetic Algorithm, Codec.

## LISTA DE FIGURAS

Figura 1 – Cronologia dos padrões de compressão de vídeo (Golston e Rao, 2006).....	13
Figura 2 – (a) Formato 4:4:4. (b) Formato 4:2:2. (c) Formato 4:2:0 do MPEG-1. (d) Formato 4:2:0 do MPEG-2 e do H.264.....	16
Figura 3 – Comparações de PSNR: (a) foto original; (b) 30.6 dB; (c) 28.3 dB (Richardson, 2003). .....	18
Figura 4 – Modelo Sistêmico da Codificação e Decodificação do H.264.....	20
Figura 5 – Perfis e ferramentas do padrão H.264/AVC. ....	24
Figura 6 – Restrições de Nível da codificação do H.264 (ITU-T, 2007).....	26
Figura 7 – Modos de predição Intra para blocos 4x4.....	29
Figura 8 – Modos de predição Intra para blocos 16x16 e 8x8. ....	30
Figura 9 – (a) Tipo 0 (16x16). (b) Tipo 1 (16x8). (c) Tipo 2 (8x16). (d) Tipo 3 (8x8). (e) Variações de SubMBs para o tipo 3.....	32
Figura 10 – Organização hierárquica da camada de codificação de vídeo.....	37
Figura 11 – Classes de complexidade de problemas de decisão. ....	38
Figura 12 – Arquitetura Sistêmica do SMC.....	54
Figura 13 – GOP - Group Of Picture (Golston and Rao, 2006).....	55
Figura 14 – Modelo Computacional do SMC. ....	68
Figura 15 – Comportamento das variáveis de decisão à medida que o peso de BR aumenta na FO.....	73
Figura 16 – Comportamento das variáveis de decisão à medida que o peso de FR aumenta na FO. ....	76
Figura 17 – Comportamento das variáveis de decisão à medida que o peso de QI aumenta na FO. ....	77
Figura 18 – Comportamento das variáveis de decisão à medida que o peso de QP aumenta na FO.....	78

Figura 19 – Comportamento das variáveis de decisão à medida que o peso de QB aumenta na FO.....	80
Figura 20 – Comportamento das variáveis de decisão à medida que o peso de PF aumenta na FO. ....	82
Figura 21 – Algoritmo SMC aplicado ao Vídeo V02.....	90
Figura 22 – Resultados da codificação do vídeo V02 quanto ao PSNR e a compressão. ....	90
Figura 23 – Algoritmo SMC aplicado ao Vídeo V10.....	91
Figura 24 – Resultados da codificação de V10 em termos de PSNR e Compressão. ....	91
Figura 25 – Comparação entre os tamanhos finais dos vídeos codificados pelo CODCE H.264 a partir da Solução Inicial e a da Solução do SMC. ....	93
Figura 26 – PSNRs da Solução Inicial e do SMC. ....	94
Figura 27 – BT com variação do nbmax aplicada ao vídeo V02.....	97
Figura 28 – BT com variação do tamanho da Lista Tabu aplicada a V02.....	98

## LISTA DE TABELAS

Tabela 1 – Tipos de macroblocos permitidos para cada tipo de <i>slices</i> .....	27
Tabela 2 – Tempo de busca em função do tamanho do problema.....	40
Tabela 3 – Algumas aplicações da Busca Tabu.....	41
Tabela 4 – Algoritmo clássico da Busca Tabu.....	44
Tabela 5 – <i>Template</i> de um algoritmo Evolucionário.....	46
Tabela 6 – <i>Crossover</i> de Um Ponto.....	50
Tabela 7 – Variação do <i>bit rate</i> de acordo com o nível e perfil do H.264. ....	63
Tabela 8 – máximo <i>frame rate</i> de acordo com o nível do H.264. ....	64
Tabela 9 – <i>Crossover</i> aritmético e crossover de um ponto.....	71
Tabela 10 – Médias obtidas à medida que o peso de BR aumenta na FO.....	75
Tabela 11 – Médias obtidas à medida que o peso de FR aumenta na FO.....	76
Tabela 12 – Médias obtidas à medida que o peso de QI aumenta na FO.....	78
Tabela 13 – Médias obtidas à medida que o peso de QP aumenta na FO.....	79
Tabela 14 – Médias obtidas à medida que o peso de QB aumenta na FO.....	81
Tabela 15 – Médias obtidas à medida que o peso de PF aumenta na FO.....	82
Tabela 16 – Vídeos da 1ª fase dos experimentos.....	84
Tabela 17 – Vídeos codificados no padrão H.264. ....	86
Tabela 18 – Médias e desvios-padrão dos algoritmos BT, AG e Híbrido. ....	94
Tabela 19 –BT com variação do nbmax aplicada ao vídeo V02.....	96
Tabela 20 – BT com variação do tamanho da lista tabu aplicada ao V02. ....	97

## LISTA DE ABREVIATURAS

AAC	Advanced Audio Coding / Codificação de Áudio Avançada
AG	Algoritmos Genéticos
ANSI	American National Standards Institute
BT	Busca Tabu
CA	Critério de Aspiração
CABAC	Context Adaptive Binary Arithmetic Coding / Codificação Aritmética Binária Adaptável ao Contexto
CAVLC	Context Adaptive Variable-Length Coding / Codificação de Comprimento Variável Adaptável ao Contexto
CODEC	Codificação e decodificação de sinais fonte
DCT	Discrete Cosine Transform / Transformada Discreta de Cosseno
DigConv	Projeto Plataforma de Convergência Digital IPTV/TV Digital
ITU	International Telecommunication Union
LC	Low Complexity
LT	Lista Tabu
MAD	Mean Absolute Difference / Média das diferenças absolutas
MPEG	Moving Picture Expert Group
NAL	Network AbstractionLayer in H.264/MPEG-4 (Advanced Video Coding)
OTD	Otimização Taxa Distorção
PSNR	Peak Signal to Noise Ratio / Razão entre pico de sinal e ruído
SAD	Sum of Absolute Differences / Soma das diferenças absolutas
SAE	Sum of Absolute Errors
SBTVD	Sistema Brasileiro de Televisão Digital
SD	Standard Definition / Definição Padrão
VM	Vetor de Movimento
YUV	Luminance/chrominance system
HD	High Definition / Alta definição

## SUMÁRIO

1	INTRODUÇÃO .....	11
2	CODIFICAÇÃO DE VÍDEO .....	11
2.1	Histórico da Codificação de Vídeo.....	13
2.2	Sinal de Vídeo Digital e Espaço de Cores .....	14
2.3	Estimativa da Qualidade de Vídeo .....	17
2.3.1	Métricas Objetivas – PSNR x MSE.....	17
2.3.2	HVS – <i>Human Visual System</i> .....	18
2.3.3	Conjunto de Parâmetros Objetivos .....	19
2.3.4	Métricas Baseadas em Referências.....	19
2.3.5	Métricas Baseadas em Semi Referências .....	19
2.3.6	Métricas Baseadas em Livres Referências .....	19
3	PADRÃO H.264/AVC. ....	20
3.1	Redundância de Dados .....	22
3.1.1	Redundância Espacial.....	22
3.1.2	Redundância Temporal.....	22
3.1.3	Redundância Entrópica.....	23
3.2	Perfis.....	24
3.3	Níveis.....	25
3.4	Macrobloco .....	26
3.5	Predição .....	28

3.5.1	Predição Intra.....	29
3.5.2	Predição Inter.....	31
3.6	Codificação do Erro Residual.....	32
3.7	Transformação .....	33
3.8	Quantização .....	34
3.9	Filtro Deblocking.....	34
3.10	Reordenação e Codificação de Entropia.....	35
3.11	Controle de Taxa no H.264/AVC .....	36
3.12	Organização dos Dados .....	37
4	METAHEURÍSTICAS .....	38
5	BUSCA TABU .....	40
6	ALGORITMO GENÉTICO .....	45
6.1	Codificação .....	46
6.2	Inicialização .....	47
6.3	Avaliação .....	47
6.4	Operador de Seleção .....	48
6.4.1	Seleção por Roleta .....	48
6.4.2	Seleção por Torneio .....	48
6.4.3	Seleção por Amostragem Estocástica.....	49
6.5	Operador de Cruzamento .....	49
6.5.1	Cruzamento de Um Ponto.....	49

6.5.2	Cruzamento Aritmético .....	50
6.6	Operador de Mutação .....	50
6.6.1	Mutação Uniforme.....	51
6.7	Operador de Elitismo.....	51
6.8	Parâmetros Genéticos .....	51
7	SMC – SIMULADOR DE METAHEURÍSTICAS APLICADO A UM CODEC ...	52
7.1	Arquitetura Sistêmica do SMC.....	53
7.1.1	Parâmetro de configuração de <i>frames</i> tipo B.....	54
7.1.2	Parâmetro <i>Bit Rate</i> .....	55
7.1.3	Parâmetros de quantização de <i>frames</i> do tipo I, P e B .....	56
7.1.4	Parâmetro <i>Frame Rate</i> .....	57
7.2	Formulação do Problema.....	58
7.2.1	Formulação Matemática .....	59
7.2.2	Solução Não Tendenciosa .....	65
7.3	Arquitetura Híbrida do Modelo SMC.....	66
7.3.1	Módulo da Busca Tabu.....	69
7.3.2	Módulo do Algoritmo Genético .....	70
8	VALIDAÇÃO DO MODELO.....	72
8.1	Primeira Etapa dos Experimentos de Validação do Modelo .....	72
9	EXPERIMENTOS UTILIZANDO O CODEC H.264.....	83
9.1	Primeira Etapa de Experimentos .....	83



9.1.1	Resultados do Primeiro Experimento .....	85
9.2	Segunda Etapa de Experimentos .....	95
9.2.1	Resultados do Segundo Experimento .....	96
10	CONCLUSÃO .....	99
10.1	Trabalhos Futuros .....	101
	REFERÊNCIAS BIBLIOGRÁFICAS .....	102
	Glossário .....	108



## 1 INTRODUÇÃO

No ano de 2005, o governo brasileiro deu início ao projeto SBTVD (ABNT, 2007) onde houve a participação de diversas universidades e centros de pesquisa. O projeto SBTVD aborda todas as partes que compõem um sistema de TV Digital e inovações que objetivaram ajustar a implantação da TV Digital à realidade dos brasileiros. Para isso o governo determinou requisitos básicos para o SBTVD dentre os quais podemos citar: baixo custo na produção dos terminais de acesso ou *set-top boxes*, robustez na recepção do sinal, flexibilidade e capacidade de evolução, interatividade dentre outros, com a finalidade de promover a inclusão digital no país. Para alcançar seu objetivo de promover a inclusão digital, o governo brasileiro determinou que os terminais de acesso à TV Digital Brasileira tivessem custos baixos para que pudessem ser acessíveis à população de baixa renda.

A TV Digital Brasileira adotou como padrão de codificação de vídeo para o SBTVD, o padrão H.264/MPEG-4 AVC, também conhecido apenas como H.264. Este padrão contém inúmeros parâmetros de configuração que o deixam com alta flexibilidade e afeta enormemente o seu desempenho. Devido a isso, um codificador H.264 configurado inadequadamente pode apresentar um desempenho muito inferior à sua real capacidade.

O H.264/MPEG-4 AVC apresenta um desempenho largamente superior aos seus antecessores como MPEG-2, H.263 e MPEG-4 parte 2. Essa melhoria de desempenho é alcançada graças ao emprego de técnicas mais sofisticadas que requerem maior custo computacional, por isso a configuração de seus parâmetros no processo de codificação/decodificação de vídeo é essencial para o seu bom desempenho.

## 2 CODIFICAÇÃO DE VÍDEO

Nos últimos anos, a demanda por produtos de vídeo digital tem testemunhado uma explosão. Alguns exemplos de aplicações populares que utilizam vídeo digital são comunicação de vídeo, segurança e vigilância, automação industrial, e a maior

de todas que é a indústria do entretenimento, que inclui DVD, HD TV, internet, câmeras digitais, monitores de vídeo de alta definição (LCD, Plasma, etc.) e muitos outros. Para poder manipular estes vídeos para que possam ser transmitidos com eficiência é necessário geralmente fazer a compressão do vídeo. Essa compressão tem como objetivo a redução do número de bits necessários para representar um vídeo digital sem que tenha perda de informações significativas. Isto é feito através da exploração de uma propriedade existente nas sequencias de vídeo: a redundância, que faz com que grande parte dos dados necessários para representar um vídeo digital seja supérflua (da Silva, 2007).

Algoritmos de compressão e descompressão de vídeo, os chamados CODECs, têm sido continuamente aperfeiçoados durante a última década para atender as demandas de mercado. Um dos CODECs mais recentes é o padrão H.264/MPEG-4 AVC o qual faz parte da terceira geração de tecnologia de compressão de vídeo. Contudo, a escolha do CODEC correto e a otimização em tempo-real de sua implementação para uma aplicação específica, continuam sendo um árduo desafio nos dias de hoje. O projeto ótimo de um CODEC significa obter compressão de dados com ótima eficiência usando recurso computacional limitado, o que se resume numa árdua tarefa (Golston e Rao, 2006).

A codificação de vídeo de um sistema de TV Digital é utilizada para reduzir a taxa de *bits* necessária para a transmissão de sinal de vídeo (CPqD, 2006a, 2006b). Este subsistema está contido no CODEC (Codificação e decodificação de Sinais Fonte).

O codificador recebe como entrada um sinal de vídeo digital não comprimido produzido na etapa de produção de conteúdo, realiza sua compressão, e gera como saída um fluxo elementar de informação de vídeo codificado que é fornecido à camada de transporte do sistema, conforme pode ser visto na arquitetura sistêmica do modelo representado pela Figura 12.

No Sistema Brasileiro de televisão Digital, o canal de transmissão adotado é o mesmo dos sistemas analógicos e tem uma largura de faixa de 6 MHz centrado em frequência das faixas VHF e UHF (CPqD, 2006a, 2006b).

A codificação de sinais fonte trata a questão da compressão dos sinais de áudio e vídeo transportados pelo sistema e dos métodos de transporte de dados. A codificação de sinais fonte é um dos principais viabilizadores da TV Digital, dada as taxas de *bits* relativamente elevadas, demandadas para transmissão destes sinais, e a necessidade de se estruturar os dados que a tecnologia permite transmitir (CPqD, 2006a, 2006b).

A taxa de *bits* necessária para representar um sinal de vídeo digitalizado sem compressão pode chegar aos valores de 270 Mbit/s e 1.5 Gbit/S para os sinais de SDTV e HDTV, respectivamente (CPqD, 2006a, 2006b) enquanto que a taxa de *bits* disponibilizada pelos atuais sistemas de transmissão de TV Digital está em torno de 19 Mbits/s quando se considera um canal de 6MHz (CPqD, 2006a, 2006b).

## 2.1 Histórico da Codificação de Vídeo

A Figura 1 mostra a cronologia dos diversos padrões de CODEC de vídeo que surgiram desde 1984 até 2012.

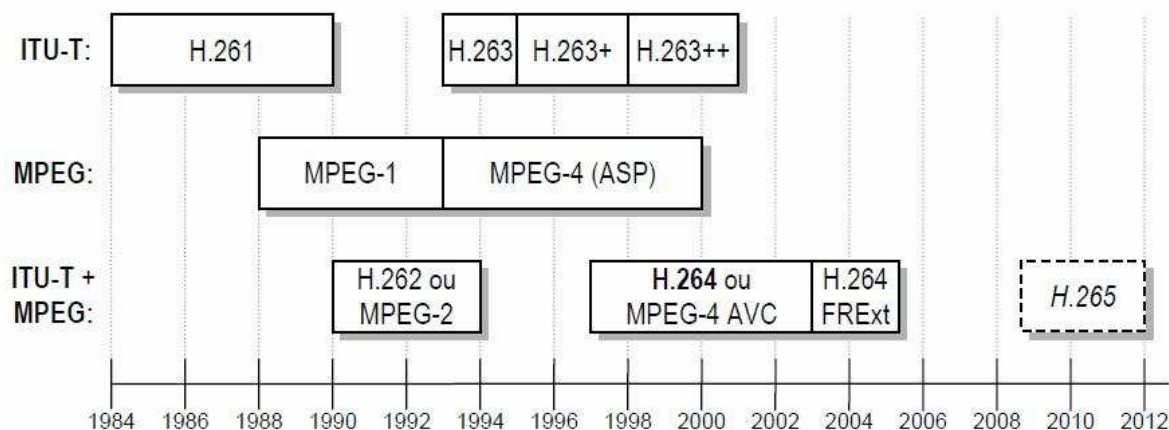


Figura 1 – Cronologia dos padrões de compressão de vídeo (Golston e Rao, 2006)

Existem duas principais organizações que conduzem a padronização dos CODECs de vídeo. A primeira é a ITU-T (*International Telecommunication Union*) focada em aplicações de telecomunicações e responsável pela criação dos padrões H.26x, que incluem o H.261, H.262, H.263 e H.264. Os primeiros padrões da ITU-T foram criados para diminuir o *bit rate* da telefonia de vídeo. A segunda é a ISO/IEC (*Moving Picture Expert Group - MPEG*) que é mais focada em aplicações para o

consumidor e tem definido o padrão MPEG que é o padrão usado para compressão de imagem em movimento e inclui os padrões MPEG-1, MPEG-2 e MPEG-4 (Manoel, 2007; Golston e Rao, 2006).

Em algumas ocasiões as duas organizações trabalham juntas e formam a *Join Video Team (JVT)*, empresa focada em definir o padrão H.264 Parte 10 ou MPEG-4 *Advanced Video Coding (AVC)* dentro da família MPEG.

Padrões de vídeo tem sido um fator crítico ao longo dos anos devido à adoção generalizada da tecnologia de CODEC. Consumidores procuram produtos baseados em padrões acessíveis devido a economias de escala. A indústria está disposta a investir em padrões que garantam a interoperabilidade entre fabricantes. Provedores de conteúdo são atraídos para padrões dados a longa vida e ampla demanda que seus conteúdos veriam. Enquanto quase todos os padrões de vídeo são direcionados a poucas aplicações específicas, eles são muitas vezes aproveitados em outras aplicações quando conseguem ser bem adaptados a elas a fim de se tirar vantagens (Golston e Rao, 2006).

## **2.2 Sinal de Vídeo Digital e Espaço de Cores**

O passo inicial em um processamento de vídeo são a amostragem temporal, a amostragem espacial e o espaço de cores.

Todo sinal de vídeo consiste de uma sequência de quadros amostrados temporalmente, que quando reproduzidos, geram a ilusão de movimento. Esses quadros passam também por uma amostragem espacial, na qual cada amostra representa uma intensidade de brilho ou de cor, em outras palavras podemos dizer que a amostragem em domínio temporal fornece um número de imagens por segundo de acordo com a taxa de *frame* e a amostragem em domínio espacial fornece um número de pontos chamados *pixels* em cada imagem de acordo com a resolução da imagem. A amostragem de cor se refere ao espaço de cor e ao número de *bits* usados para representar a cor de um único *pixel* (Manoel, 2007; Ries, 2008).

No sistema óptico humano, a percepção das cores se deve às células fotossensíveis chamadas cones, que se dividem em três tipos de fotorreceptores

cujas respostas variam de acordo com o comprimento de ondas. Devido a isso, para que seja possível criar todas as cores que podemos visualizar, são necessários três sinais de cor. Sabendo-se que o sistema de visão humano é mais sensível à luminosidade do que à cor então podemos minimizar o tamanho da banda armazenando mais detalhes de luminosidade e menos detalhes de cor. Numa distância de visão normal, não há percepção de perda nos detalhes de cor em decorrência de baixas taxas de transmissão (Manoel, 2007; Ries, 2008).

Em todos os tipos de TVs e monitores coloridos, sejam baseados em LCD, plasma ou CRT, as cores são formadas a partir da combinação de três cores “primárias”: vermelho, verde e azul que dão origem ao sistema denominado RGB (*Red-Green-Blue*). Apesar de ser um espaço de cor adequado para os dispositivos de visualização, essa representação não é tão propícia para a codificação de vídeo, já que existe uma alta correlação entre os três sinais, resultando em redundâncias (Manoel, 2007; Ries, 2008).

Através de transformações lineares e não lineares, pode-se representar as cores em outros diferentes espaços de cor, tal como Y’CbCr, que é o mais apropriado para a codificação de vídeo devido as seguintes características (Manoel, 2007):

- Não apresenta tão alta redundância quanto o RGB;
- Separa o sinal representativo de intensidade de luz (luma) dos dois sinais responsáveis pela colorização (crominância ou croma).

A conversão de RGB para Y’CbCr é padronizada pelo ITU-R BT.601. Neste padrão o sinal é dividido em um componente de luminância (denotada como Y, conhecida como “luma”) e dois componentes de diferença de cor chamados de crominância (croma) e denotados por U e V ou Cb e Cr respectivamente (Manoel, 2007; Ries, 2008).

Devido ao sistema óptico humano ser mais sensível à intensidade de luz que às cores, pode-se utilizar uma resolução menor na amostragem dos sinais croma quase sem prejuízos à qualidade subjetiva das imagens. Existe uma nomenclatura padrão para as diferentes formas de subamostragem comumente utilizadas em codificação de vídeo (Manoel, 2007):

- No formato 4:4:4, não há subamostragem, ou seja, para cada amostra  $Y'$ , existe uma amostra Cb e uma amostra Cr correspondentes;
- No formato 4:2:2, a resolução dos dois sinais croma é reduzida pela metade horizontalmente;
- No formato 4:2:0, a resolução dos dois sinais croma é reduzida pela metade tanto horizontalmente quanto verticalmente, ou seja, para cada quatro amostras  $Y'$  existe uma amostra Cb e uma amostra Cr.

A posição das amostras está ilustrada na Figura 2. Para que uma imagem com subamostragem croma seja mostrada corretamente, as amostras cromas devem ser interpoladas nas posições das amostras luma, antes da conversão para o espaço de cores RGB (Manoel, 2007).

A Figura 2 mostra a posição relativa das amostras luma, Croma Cb e Croma Cr, em diferentes formatos de subamostragem.

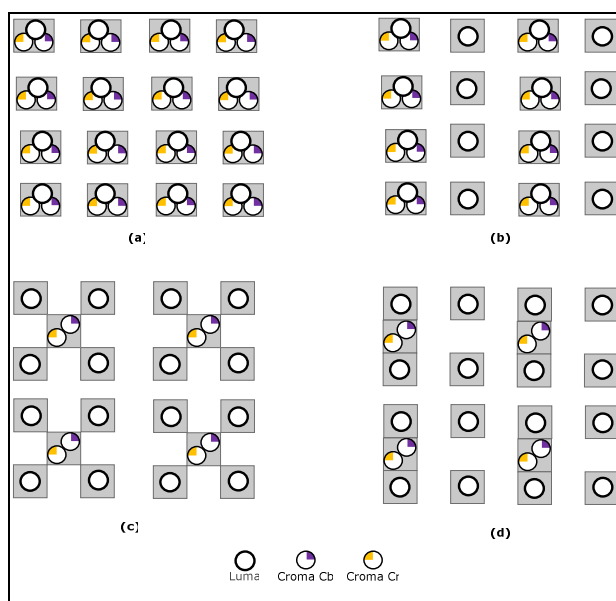


Figura 2 – (a) Formato 4:4:4. (b) Formato 4:2:2. (c) Formato 4:2:0 do MPEG-1. (d) Formato 4:2:0 do MPEG-2 e do H.264.



## 2.3 Estimativa da Qualidade de Vídeo

Até hoje vários métodos foram propostos para a estimação da qualidade de vídeo. Esses métodos podem ser classificados em métodos objetivos e métodos subjetivos (Pinson e Wolf, 2003; Wolf e Pinson, 2007; Ries, 2008).

### 2.3.1 Métricas Objetivas – PSNR x MSE

As primeiras medidas de distorção da qualidade de vídeo são o MSE (Mean Square Error) e o PSNR (*Peak Signal to Noise Ratio*) (Winkler e Mohandas, 2008). Ambos refletem pobremente a qualidade de vídeo subjetiva. Contudo, o PSNR continua sendo amplamente usado como método de referência para comparar performance de algoritmos de codificação de vídeo.

Quanto maior o PSNR, melhor a qualidade da imagem. O PSNR é geralmente calculado sobre as amostras luma e é dado em decibéis (dB) através da seguinte expressão (Wang *et al*, 2003):

(1)

$$MSE = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2 \quad (1)$$

(2)

$$PSNR = 10 \log_{10} \frac{L^2}{MSE} \quad (2)$$

Onde  $N$  é o número de *pixels* da imagem ou sinal e  $x_i$  e  $y_i$  são os *pixels* da posição  $i$  dos sinais original e distorcido respectivamente.  $L$  é a faixa dinâmica dos valores do *pixel*.

A Figura 3 mostra três imagens, onde a imagem (a) é o quadro original e as imagens (b) e (c) são versões com menor qualidade originadas a partir da codificação e decodificação do quadro original. A imagem (b) tem um PSNR de 30,6 dB enquanto a imagem (c) tem um PSNR de 28,3 dB, sendo portanto, uma imagem de pior qualidade em relação às anteriores.



Figura 3 – Comparações de PSNR: (a) foto original; (b) 30.6 dB; (c) 28.3 dB (Richardson, 2003).

A principal limitação em relação ao PSNR é a necessidade da imagem original para se fazer o cálculo em relação à degradação de um vídeo.

### 2.3.2 HVS – *Human Visual System*

Métricas baseadas no sistema visual humano, as chamadas HVS (Human Visual System). As métricas assumem que os múltiplos canais de mediação da percepção visual são independentes entre si. Contudo, achados da neurociência e experimentos psicofísicos estabelecem que exista uma interação entre os canais e tais interações são importantes para o mascaramento visual. Assim, em um futuro próximo, os melhores modelos de HVS, para avaliação de qualidade confiável, são esperados. A principal desvantagem desses modelos HVS é sua alta complexidade computacional.

### 2.3.3 Conjunto de Parâmetros Objetivos

Métricas baseadas em um conjunto de parâmetros objetivos provêm um bom *trade-off* entre eficácia e complexidade. O conjunto de parâmetros consiste de parâmetros objetivos sensíveis a qualidade. Esta abordagem é muito apropriada para estimação de qualidade em cenários com uso definido, conteúdo e condições de serviço de vídeo.

### 2.3.4 Métricas Baseadas em Referências

São métricas calculadas a partir da diferença computacional entre as sequências de vídeo original e degradada. As diferenças podem ser usadas para calcular medidas de distorção comparativas, as quais têm uma baixa correlação com a perda percebida mas, são fáceis de extrair. A referência é necessária na entrada fortemente restrita a sua aplicabilidade.

### 2.3.5 Métricas Baseadas em Semi Referências

São medidas obtidas através da computação de parâmetros de uma imagem degradada e compara esses parâmetros com os mesmos parâmetros calculados a partir da imagem de referência. Indicadores de qualidade podem ser obtidos comparando parâmetros calculados separadamente da imagem codificada e as imagens de referência. Esses parâmetros podem ser distribuídos na rede a um baixo *bit rate* para ser usado quando o sinal de referência completo não estiver disponível.

### 2.3.6 Métricas Baseadas em Livres Referências

Essas métricas não requerem nenhum conhecimento da fonte de vídeo original. Essas métricas encontram uma dificuldade básica em dizer distorções à parte a partir do conteúdo regular, o qual é algo que os humanos fazem bem usando sua experiência. A sua grande vantagem é a sua versatilidade e flexibilidade.

### 3 PADRÃO H.264/AVC.

A Figura 4 mostra os principais processos que compõem o padrão de codificação de vídeo H.264/AVC usado no SBTVD.

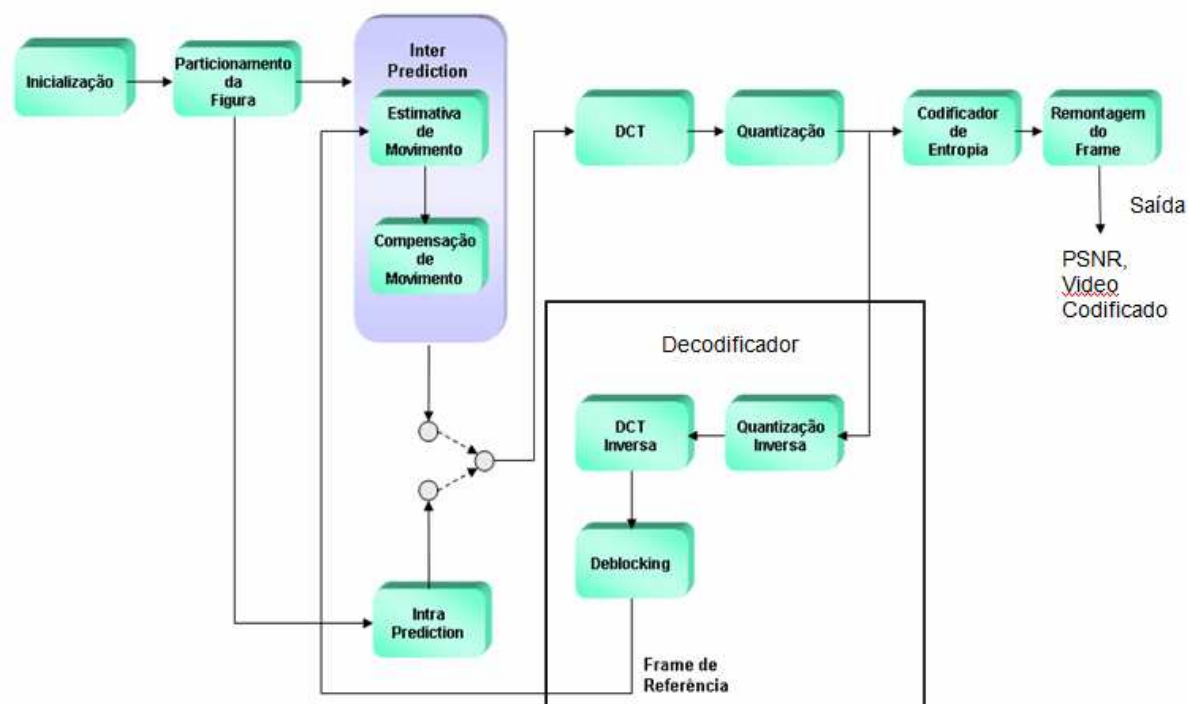


Figura 4 – Modelo Sistêmico da Codificação e Decodificação do H.264

O padrão de codificação usado no SBTVD (ABNT, 2007), o H.264/MPEG-4 AVC, abrange uma enorme faixa de aplicações que englobam desde vídeos a baixas taxas (vídeo para celulares) até altas taxas de transmissão (difusão de TV), bem como diversas resoluções espaciais (quantidades de *pixels* por imagem) e temporais (taxa de quadros ou imagens por segundo).

O padrão H.264/AVC atingiu seu objetivo de alcançar as mais elevadas taxas de processamento dentre todos os padrões existentes. Mas para isso, foi necessário um grande aumento na complexidade computacional das operações dos CODECs que seguem o padrão H.264/AVC em relação aos demais padrões disponíveis na atualidade. Este aumento de complexidade impede, pelo menos na tecnologia atual, a utilização de CODECs H.264/AVC implementados em software quando as resoluções são elevadas e/ou quando se deseja tempo real, com 30 quadros por segundo, por exemplo. A intratabilidade do problema via software somado ao enorme interesse comercial que reside neste padrão, têm impulsionado equipes de

pesquisa e desenvolvimento ao redor do mundo a tratarem deste tema visando otimizações algorítmicas e/ou implementações em hardware para que os requisitos das aplicações sejam atendidos (da Silva, 2007).

Considerando a diversidade de conteúdos a serem transmitidos, tais como: apresentação de telejornal, filmes, programas esportivos, etc.; a escolha adequada desses parâmetros se torna ainda mais crítica, pois, devido às suas características, esses conteúdos em geral requerem configurações de codificador e complexidades computacionais bastante distintas para uma mesma qualidade de vídeo pré-definida (IME *et al*, 2010).

A etapa mais custosa do codificador do H.264 é a estimação de movimento, responsável pela busca de descolamentos entre quadros, com um gasto de aproximadamente 90% do tempo total de codificação. Por causa disso, qualquer alteração feita no codificador que implique aumento no tempo gasto para a estimação de movimento terá um impacto final considerável no tempo de execução da codificação. Este é um caso de aumento da complexidade mais perceptível que pode ser provocado pelo aumento no número de partições ou formas de se dividir o macrobloco.

Com tudo isso, a compressão de vídeo e o padrão H.264/AVC são assuntos que estão sendo amplamente estudados por pesquisadores do meio acadêmico e da indústria e onde muitas contribuições significativas podem ainda serem dadas.

A compressão de vídeo é uma técnica que faz parte do grupo mais geral de tecnologias de compressão de dados, que se classificam em compressão sem perdas e compressão com perdas. No primeiro caso, após o processo de compressão/descompressão, o sinal reconstituído é idêntico ao original, enquanto no segundo caso há uma degradação desse sinal, denominada distorção (Manoel, 2007).

A compressão sem perdas consegue, geralmente, compactar dados numa ordem de grandeza de duas a três vezes. Já na compressão com perdas, a taxa resultante depende apenas da distorção admitida – as tecnologias de compressão mais recentes podem comprimir para uma taxa até 100 vezes menor do que a taxa

original, com uma distorção ainda aceitável. O tipo de compressão do padrão H.264 é a compressão com perdas (Manoel, 2007).

### 3.1 Redundância de Dados

O objetivo da compressão de vídeo é eliminar as informações que são redundantes em um vídeo digital, e assim, diminuir o número de *bits* necessários para a sua representação. Os padrões de compressão de vídeo exploram justamente uma característica dos vídeos digitais que são os vários tipos de redundância. Existem basicamente três tipos de redundâncias que são exploradas na compressão de vídeos: Espacial, Temporal e Entrópica (da Silva, 2007).

#### 3.1.1 Redundância Espacial

A redundância espacial, também chamada de redundância intra-quadro ou redundância interpixel, advém da correlação existente entre os pixels espacialmente distribuídos em um quadro. Esta correlação pode ser percebida tanto no domínio espacial quanto no domínio das frequências. No domínio espacial, pode-se observar *pixels* vizinhos em um quadro que tendem a possuir valores semelhantes, neste caso, a redundância pode ser reduzida através de uma operação chamada Codificação Intra-quadro. No domínio das frequências a operação realizada para reduzir a redundância espacial é chamada de Quantização. Para aplicar a Quantização, as informações da imagem devem ser transformadas do domínio espacial para o domínio das frequências. A quantização é uma divisão inteira dos coeficientes gerados pela transformação e reduz grande parte dos coeficientes à zero. Esta operação gera perdas no processo de codificação de vídeo (da Silva, 2007).

#### 3.1.2 Redundância Temporal

Na verdade, a redundância temporal poderia ser classificada como apenas mais uma dimensão da redundância espacial, como faz Gonzales e Woods (2003).

Muitos blocos de pixels simplesmente não mudam de valor de um quadro para outro em um vídeo, como por exemplo, em um fundo que não foi alterado de um quadro para outro. Outros pixels apresentam uma pequena variação de valores causada por exemplo, por uma variação de iluminação. Por fim, também é possível que o bloco de pixels simplesmente tenha se deslocado de um quadro para o outro, como por exemplo, em um movimento de um objeto em uma cena. Todos os padrões atuais de codificação de vídeo visam reduzir a redundância temporal. A exploração eficiente da redundância temporal conduz a elevadas taxas de compressão, o que é fundamental para o sucesso dos codificadores de vídeo (da Silva, 2007).

### 3.1.3 Redundância Entrópica

A redundância entrópica está relacionada com as probabilidades de ocorrência dos símbolos codificados. A entropia é a medida da quantidade média de informação transmitida por símbolo do vídeo. A quantidade de informação nova transmitida por um símbolo diminui à medida que a probabilidade de ocorrência deste símbolo aumenta. Então, os codificadores que exploram a redundância entrópica têm por objetivo transmitir o máximo de informação possível por símbolo codificado e deste modo, representar mais informações com um número menor de *bits*. A codificação de entropia, como é chamada, utiliza diferentes técnicas e algoritmos de compressão sem perdas para atingir este objetivo (da Silva, 2007).

Além destes três tipos de redundância, existe também outro tipo que é explorada pelo padrão H.264/AVC e que diz respeito às características do sistema visual humano, que fazem com que não consigamos captar alguns tipos de informações na imagem. Algumas informações da imagem como o brilho, por exemplo, são mais importantes para o sistema visual humano do que outras, como a crominância. Este tipo de redundância de informação é chamado de redundância psicovisual e é explorada pelo padrão H.264/AVC na subamostragem de cores utilizadas pelo vídeo digital, pois utilizando um subespaço de amostragem do tipo YCbCr 4:2:0 então já existe uma eliminação das informações de cor do vídeo, fazendo com que este já esteja representado com uma compressão referente a este tipo de redundância (da Silva, 2007).

### 3.2 Perfis

O termo perfil estabelece a complexidade da codificação do vídeo. Os perfis definem um subconjunto da sintaxe do fluxo de *bits* que são adotados visando, em geral, determinadas classes de aplicações (ABNT NBR 15602-1, 2007).

O padrão H.264/AVC contém um grande conjunto de ferramentas de codificação de vídeo, contudo, não é necessário a implementação de todas elas para as aplicações de codificação. Desse modo, o padrão define subconjuntos de ferramentas de codificação que são implementadas de acordo com a classe da aplicação. Um Codificador/Decodificador de vídeo pode escolher implementar somente um subconjunto de ferramentas e a esse subconjunto dá-se o nome de Perfil. As estruturas desses perfis podem ser vistas na Figura 5.

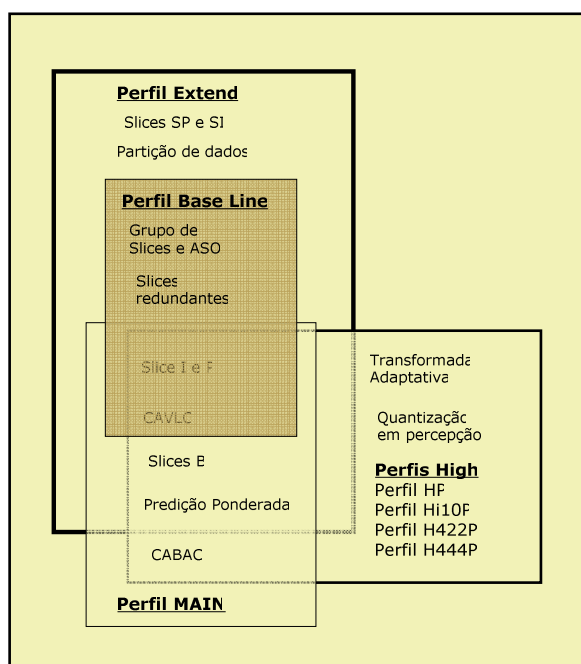


Figura 5 – Perfis e ferramentas do padrão H.264/AVC.

O perfil *Baseline* é direcionado a aplicações como videotelefonia, videoconferência e comunicações sem fio. Ele suporta codificação intra-quadro e inter-quadro (usando *slices* I e P respectivamente) além de uma codificação de entropia com códigos de comprimento variável de contexto adaptativo (*Context-Adaptive Variable-Length - CAVLC*) (Alencar, 2007; ABNT NBR 15602-1, 2007).

O perfil *Main* suporta *slices* do tipo I, P ou B. Suporta vídeo entrelaçado, codificação inter-quadros usando *slices* B, codificação inter-quadros usando



predição ponderada e codificação aritmética de contexto adaptativo (*Context-Based Arithmetic Coding – CABAC*). É usado em radiodifusão televisiva e armazenamento de vídeo (Alencar, 2007; ABNT NBR 15602-1, 2007).

O perfil *Extended* não suporta vídeo entrelaçado nem CABAC, mas oferece suporte a métodos de permutação de diferentes fluxos de *bits* codificados além de resistência e recuperação de erros aprimorados. Usado principalmente em aplicações de *streaming* de mídias (Alencar, 2007).

Estes três perfis inicialmente propostos pelo padrão H.264/AVC não incluíam suporte para vídeos com qualidade mais elevada, como as necessárias em ambientes profissionais. Para responder às exigências deste tipo de aplicação, uma continuação do projeto JVT (*Join vídeo Team*) foi realizada para adicionar novas extensões para as capacidades do padrão original. Estas extensões foram chamadas de extensões para alcance de fidelidade (*Fidelity Range Extensions – FExt*) e incluem (Alencar, 2007):

- Suporte ao dimensionamento adaptativo do tamanho dos blocos para a transformada espacial;
- Suporte a um modelo de adaptação perceptual das matrizes de quantização escaláveis;
- Um conjunto de 4 novos perfis conhecidos como Perfis *High* que são: *High profile* (HP), *High 10 Profile* (Hi10P), *Hi 4:2:2 Profile* (H422P), *High 4:4:4 Profile* (H444P).

### 3.3 Níveis

O termo nível estabelece para cada perfil os valores permitidos de alguns parâmetros de codificação, como, por exemplo, taxas de bits e resolução.

Os níveis acrescentam ainda mais restrições às definidas pelos perfis para garantir a interoperabilidade entre os diferentes decodificadores, especificando o limite de desempenho requerido para realizar a decodificação. O codificador deve

ser configurado para operar no nível especificado para cada perfil (ABNT NBR 15602-1, 2007).

A conjugação dos diferentes tipos de perfis e níveis permite a sua utilização em diferentes aplicações. A combinação perfil e nível garante a conformidade, bem como a interoperabilidade. Ou seja, uma implementação de um perfil em um nível garante que o decodificador se comporte conforme o determinado pelo padrão de codificação (ABNT NBR 15602-1, 2007).

A Figura 6 especifica as limitações impostas por cada nível da codificação de vídeo do padrão H.264.

Número do nível	Taxa máxima de processamento do macrobloco MaxMBPS MB/s	Tamanho máximo do quadro MaxFS MB	Tamanho máximo do buffer de decodificação MaxDPB (1024 bytes para 4:2:0)	Taxa máxima de vídeo MaxBR (1 000 bits/s, 1 200 bits/s, cpbBrVclFactor bits/s, or cpbBrNalFactor bits/s)	Tamanho máximo CPB (1 000 bits, 1 200 bits, cpbBrVclFactor bits, or cpbBrNalFactor bits)	Faixa vertical do componente MV MaxVm vR (uma frame samples)	Razão mínima de compressão MinCR	Número máximo de vetores de movimento por dois consecutivos MB MaxMvsPer2Mb
1	1 485	99	148,5	64	175	[- 64; + 63,75]	2	-
1b	1 485	99	148,5	128	350	[- 64; + 63,75]	2	-
1.1	3 000	396	337,5	192	500	[- 128; + 127,75]	2	-
1.2	6 000	396	891,0	384	1 000	[- 128; + 127,75]	2	-
1.3	11 880	396	891,0	768	2 000	[- 128; + 127,75]	2	-
2	11 880	396	891,0	2 000	2 000	[- 128; + 127,75]	2	-
2.1	19 800	792	1 782,0	4 000	4 000	[- 256; + 255,75]	2	-
2.2	20 250	1 620	3 037,5	4 000	4 000	[- 256; + 255,75]	2	-
3	40 500	1 620	3 037,5	10 000	10 000	[- 256; + 255,75]	2	32
3.1	108 000	3 600	6 750,0	14 000	14 000	[- 512; + 511,75]	4	16
3.2	216 000	5 120	7 680,0	20 000	20 000	[- 512; + 511,75]	4	16
4	245 760	8 192	12 288,0	20 000	25 000	[- 512; + 511,75]	4	16
4.1	245 760	8 192	12 288,0	50 000	62 500	[- 512; + 511,75]	2	16
4.2	522 240	8 704	13 056,0	50 000	62 500	[- 512; + 511,75]	2	16
5	589 824	22 080	41 400,0	135 000	135 000	[- 512; + 511,75]	2	16
5.1	983 040	36 864	69 120,0	240 000	240 000	[- 512; + 511,75]	2	16

Figura 6 – Restrições de Nível da codificação do H.264 (ITU-T, 2007).

### 3.4 Macrobloco

Em diversos padrões de codificação de vídeo, a unidade básica de codificação é o macrobloco (MB). O MB corresponde a uma matriz de 16x16 amostras luma e suas amostras croma correspondentes. Por exemplo, no formato 4:2:0, cada MB consistirá de uma matriz de 16x16 amostras luma (Y') e duas matrizes de 8x8 amostras croma (uma para Cb e outra para Cr) (Manoel, 2007).

A divisão de cada quadro em pequenos blocos de amostras é fundamental para o bom desempenho da maioria dos processos envolvidos na codificação de vídeo, como a compensação de movimento e a transformação. Como cada bloco possui características distintas, essa divisão permite que cada diferente bloco seja codificado com os parâmetros e coeficientes mais apropriados. Além disso, as amostras de blocos pequenos tendem a ser mais homogêneas, formando imagens relativamente simples, o que auxilia ainda mais no desempenho de tais processos. Especificamente no padrão H.264, os MBs podem ainda ser subdivididos em blocos menores, tanto para fins de compensação de movimento quanto para fins de transformação. Quando um bloco de 16×16 amostras luma é dividido em 4 blocos de 8×8 amostras, esses blocos são denominados submacroblocos (SubMB) (Manoel, 2007).

Os MBs são numerados sequencialmente em cada quadro e são organizados em estruturas denominadas *slices*.

Um MB pode ser codificado como:

- I (Intra) – Intra-codificado. Utiliza predição baseada em MB vizinhos do mesmo quadro.
- P (Predito) – Pode ser predito a partir da Compensação de Movimento de quadros de referência.
- B (Bi-Predito) – pode ser predito a partir de um ou dois quadros de referência.
- SI – Usado para transmissão entre duas sequencias diferentes (isto é, uma com menor taxa de *bits* que a outra) para aplicações de fluxo contínuo (*streaming*) adaptativo ou para a realização de avanço rápido de cenas (*fast-forward*).

Tabela 1 – Tipos de macroblocos permitidos para cada tipo de *slices*

Tipo de quadros	Tipos de macrobloco
I	I
P	P, I
B	B, I
SI	SI, I
SP	P, I

Fonte: ITU-T (2007).

Os *slices* também se dividem em I, P, B, SP e SI conforme pode ser visto na Tabela 1.

### 3.5 Predição

A análise estatística dos sinais de vídeo indica que existe uma alta correlação tanto entre quadros sucessivos, quanto entre os pixels dentro de cada quadro. O objetivo da compressão é representar com um número inferior de *bits* os dados originais. No caso da compressão de vídeo exploram-se as redundâncias temporais e espaciais, de forma a reduzir a representação do vídeo original com finalidade de armazenamento e/ou transmissão (ABNT NBR 15602-1, 2007).

Para se codificar imagens estáticas exploram-se apenas as redundâncias espaciais combinadas com técnicas de redução dos dados produzidos pelo primeiro processo. Entre estas técnicas, pode ser citada a codificação por código de comprimento variável. A diminuição das redundâncias espaciais pode ser realizada pelo emprego de codificação por transformadas ou sub-bandas. Um processo de codificação por transformadas, largamente utilizado pelos codificadores de imagem, baseia-se no uso da transformada inteira, geralmente combinada com a partição da imagem ou quadro em blocos. Em geral, denomina-se este tipo de codificação como codificação do tipo *intra-frame* ou Predição Intra (ABNT NBR 15602-1, 2007).

Para se codificar um vídeo, ou seja, uma sequência de imagens, pode-se aliar a um processo de redução das redundâncias espaciais um processo de redução das redundâncias temporais. Tal processo visa utilizar as diferenças entre quadros sucessivos da sequência de vídeo, com a finalidade de reduzir as redundâncias temporais. Em geral, denomina-se este tipo de codificação como codificação do tipo *inter-frame* ou Predição Inter. Técnicas de estimação de movimento e compensação de movimento podem fazer parte do processo de redução das redundâncias temporais (ABNT NBR 15602-1, 2007).

O processo de predição consiste basicamente em utilizar valores de amostras previamente decodificadas para prever os valores das amostras correntes. Esses valores são subtraídos das amostras antes de passar pela transformação de forma a reduzir ainda mais os valores a serem codificados. A predição remove redundâncias

temporais através de amostras de outros quadros, Predição Inter, ou redundâncias espaciais através de amostras do mesmo quadro, Predição Intra, (Manoel, 2007).

### 3.5.1 Predição Intra

É o modo de predição que tem por finalidade reduzir a redundância espacial da imagem, onde cada *frame* é tratado individualmente. Este processo é acionado para macroblocos do tipo I e SI (ITUT, 2007).

Neste processo as entradas a serem consideradas são amostras construídas antes do processo de Filtro antiblocagem e para modos de predição IntraNxN (onde NxN pode ser 4x4 ou 8x8) (ITUT, 2007).

Na fase de codificação, a Predição Intra gera um bloco de predição (denominado PRED), utilizando amostras vizinhas através da cópia dos pixels, de acordo com um padrão pré-estabelecido. Este bloco gerado é utilizado sobre o bloco original (denominado ORIG), de forma a encontrar uma diferença residual (Residual = ORIG – PRED). Existem diferentes padrões de geração do bloco PRED, sendo assim, cada um é gerado e calculado seu residual. Estes são então comparados para a escolha do que gera menos perdas à imagem. O bloco residual do modo eleito é enviado para a saída junto à informação do padrão escolhido (UNISINOS, 2008a).

A Predição Intra é realizada copiando-se o valor de pixels de blocos vizinhos (já codificados e reconstruídos) para o atual, de acordo com padrões pré-definidos. Para a predição em blocos 4x4 de luminância, existem 9 modos possíveis, ou seja, 9 padrões diferentes para a cópia dos pixels vizinhos, conforme ilustra a Figura 7 (UNISINOS, 2008b).

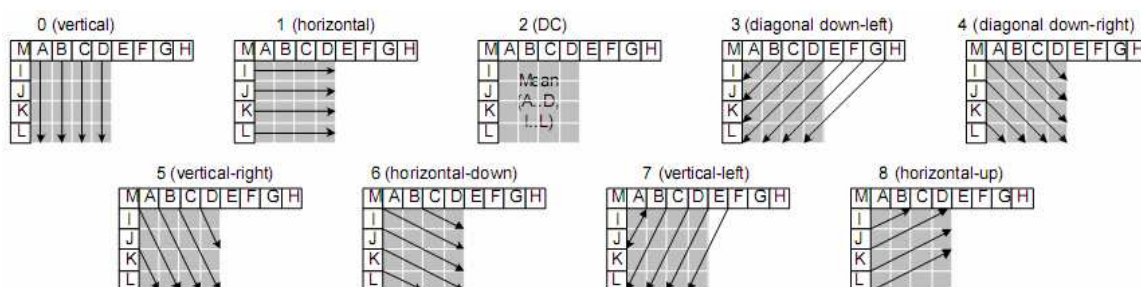


Figura 7 – Modos de predição Intra para blocos 4x4.

Para a predição em blocos luma\_16x16, existem 4 modos possíveis, conforme Figura 8. Para a predição de blocos 8x8 de componentes de crominância, os padrões são os mesmos utilizados no modo luma\_16x16. As componentes de crominância azul e vermelha (Cb e Cr) são codificadas sempre usando o mesmo modo de *intra prediction* (UNISINOS, 2008a).

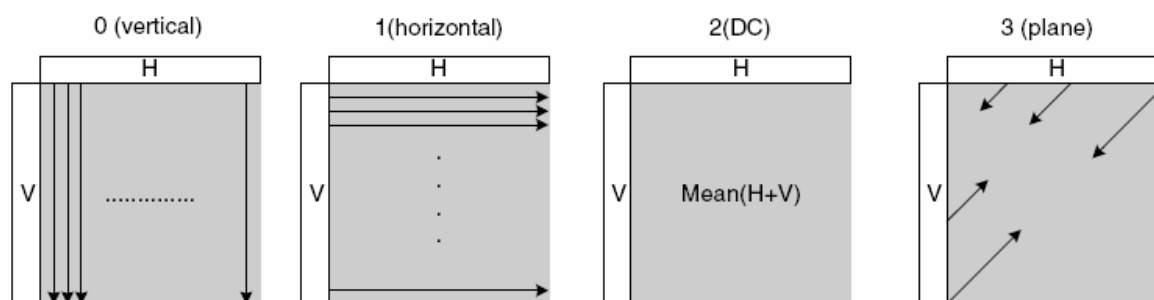


Figura 8 – Modos de predição Intra para blocos 16x16 e 8x8.

Na decodificação, através dos blocos vizinhos já decodificados (acima e à esquerda) e das informações sobre os modos selecionados, o MB é decodificado e reconstituído. É feito o cálculo inverso da Predição Intra na codificação ( $ORIG = Residual + PRED$ ) (UNISINOS, 2008a).

A decisão da predição é feita com base na comparação entre os diferentes valores de SAE (*Sum of Absolute Errors*) dos blocos e os valores de COST do macrobloco, apresentados a seguir.

Para definição do padrão a ser utilizado na Predição Intra, são aplicados todos os padrões possíveis e é escolhido aquele que apresenta o menor índice SAE (Sum of Absolute Errors – ou soma absoluta dos erros). O SAE é resultado da soma dos módulos dos pixels no bloco, como mostra a equação (3) (UNISINOS, 2008b):

$$SAE = |O_0 - P_0| + |O_1 - P_1| + \dots + |O_n - P_n| \quad (3)$$

SAE é uma medida de distorção e corresponde à soma do valor absoluto das diferenças entre a amostra original ( $O_n$ ) e sua correspondente a pós o processo de codificação ( $P_n$ ).

Quanto menor o valor do SAE, menor o erro existente entre o bloco original e o residual. Assim, o modo com menor valor de SAE é a melhor opção.

A seleção entre o modo de predição por blocos 4x4 ou 16x16 é feita através da comparação dos seus custos. Deve-se calcular o custo individual de cada bloco residual 4x4 e somar o custo total. Este valor é comparado com o custo do macrobloco. A predição que tiver o menor custo será a escolhida. A equação (4) relaciona a seleção entre os custos de predição e sua escolha (Unisinos, 2008a):

$$\text{PRED} \begin{cases} \text{COST}_{16 \times 16} \leq \text{COST}_{4 \times 4} & \rightarrow \text{Predição em macrobloco} \\ \text{COST}_{16 \times 16} > \text{COST}_{4 \times 4} & \rightarrow \text{Predição em blocos 4x4} \end{cases} \quad (4)$$

### 3.5.2 Predição Inter

A Predição Inter (ou do inglês, *inter prediction*) é o modo de predição utilizado para reduzir a redundância de informações entre quadros denominada redundância temporal. Essa predição consiste em subtrair um quadro de referência (ou quadro de predição), que seja semelhante ao quadro atual, do próprio quadro atual, para obter uma diferença entre ambos, que será codificada e transmitida. Em outras palavras temos que  $\text{DiffFrame} = \text{FrameAtual} - \text{RefFrame}$  (UNISINOS, 2008b).

O método descrito acima é muito básico e não resulta em uma compressão eficiente. Logo, é usado um método aprimorado de predição Intra, dividindo-se a figura em blocos. Este processo pode ser dividido em duas etapas: estimativa de movimento (*motion estimation*) e compensação de movimento (*motion compensation*). O processo de estimativa de movimento consiste em procurar no quadro de referência, um bloco mais semelhante possível ao bloco atual. A compensação de movimento tem como função, após escolhido o bloco de predição, subtraí-lo do bloco atual para se obter um bloco residual ( $\text{BlocoRes} = \text{BlocoAtual} - \text{BlocoRef}$ ). É calculada então a diferença de posição entre o bloco de referência e o bloco atual, o que resulta em um Vetor de Movimento (*motion vector*). O bloco residual e o vetor de movimento são então codificados e transmitidos (UNISINOS, 2008b).

Além de se poder utilizar apenas um Vetor de Movimento para o MB inteiro (MB\_16x16), o padrão H.264 também prevê a flexibilidade de se particionar o MB e realizar a compensação de movimento em blocos menores. O MB pode ser particionado em dois blocos de 16x8, dois blocos de 8x16 ou quatro blocos 8x8 (denominados partições), sendo que, nesse último caso, cada bloco 8x8 pode ser novamente particionado em dois blocos 8x4, dois blocos 4x8 ou quatro blocos 4x4 (denominados subpartições), totalizando 259 ( $3 + 4^4 = 259$ ) particionamentos possíveis, conforme mostrado na Figura 9 (Manoel, 2007).

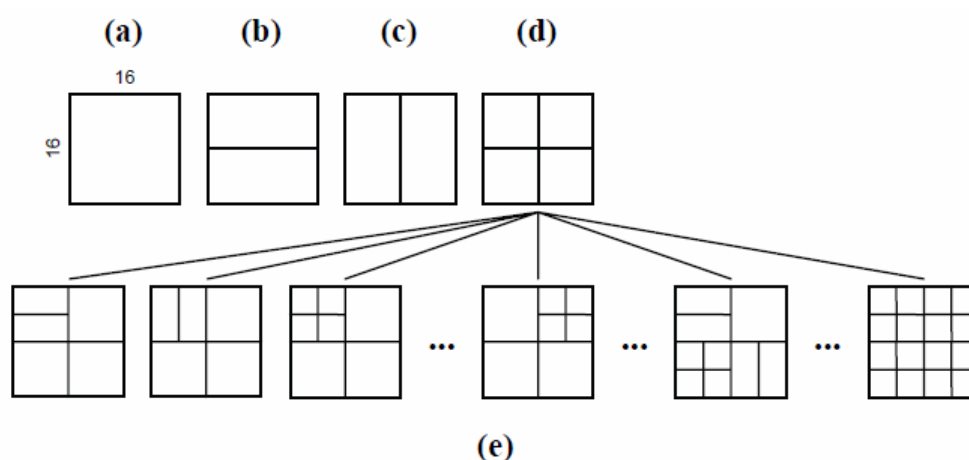


Figura 9 – (a) Tipo 0 (16x16). (b) Tipo 1 (16x8). (c) Tipo 2 (8x16). (d) Tipo 3 (8x8). (e) Variações de SubMBs para o tipo 3.

O uso de blocos menores para a compensação de movimento aumenta a possibilidade de encontrar um bloco, no quadro de referência, o mais parecido possível com o bloco atual. Isto diminui as informações no bloco residual, entretanto aumenta o número de Vetores de Movimento (uma vez que, para cada bloco, é necessário um novo vetor). Em geral, uma partição maior é melhor para áreas homogêneas da imagem, enquanto blocos menores são aconselháveis para regiões detalhadas da imagem. Uma boa escolha do tamanho das partições tem um impacto significativo no desempenho da compressão de vídeo (UNISINOS, 2008a).

### 3.6 Codificação do Erro Residual

Realizadas as predições, o que é realmente codificado é o “erro residual”, que é a diferença entre as amostras atuais e as amostras preditas. A codificação do erro residual em um MB no padrão H.264 é feita da seguinte forma:



- Cada um dos 16 blocos 4x4 luma e 4 blocos 4x4 croma são transformados por uma transformada inteira baseada no DCT.
- No caso em que o MB seja do tipo Intra\_16x16, o bloco 4x4 de coeficientes DC luma considera a transformada de *Hadamard* 4x4 (“transformada da transformada”).
- Em blocos 2x2 de coeficientes DC croma é utilizada a transformada de Hadamard 2x2.

Após a realização da transformada inteira baseada em DCT, nos blocos 4x4, os coeficientes são escalados e quantizados.

### 3.7 Transformação

As transformadas se baseiam na DCT (*Discrete Cosine Transform*) e têm como objetivo transformar os blocos de imagem (matrizes de pixels) em coeficientes. Isto faz com que grande parte da informação do bloco se concentre na área superior esquerda do mesmo, viabilizando o processo de quantização (UNISINOS, 2008a).

As transformadas utilizadas no padrão H.264 são semelhantes à DCT, porém otimizadas para trabalhar em hardware com mais eficiência utilizando apenas cálculos com valores inteiros e multiplicações e divisões múltiplas de dois (UNISINOS, 2008b).

O codificador H.264 utiliza três transformadas dependendo do tipo de dado residual que é passado que são: uma Transformada Hadamard para um bloco 4x4 de coeficientes DC da parte Luma (HCT 4x4); uma Transformada Hadamard para um bloco 2x2 de coeficientes DC da parte Chroma (HCT 2x2); e uma transformada DCT-like para todos os outros blocos 4x4 de dados residuais (UNISINOS, 2008a).

Na primeira etapa é aplicada a Transformada Forward (DCT-Like). Se o bloco transformado foi codificado em intra prediction, aplica-se a segunda etapa do processo de transformação. É transmitido um bloco com os coeficientes DC Luma para a Transformada Hadamard 4x4 e logo depois um bloco de coeficientes DC Chroma para a Transformada Hadamard 2x2 (UNISINOS, 2008a).

Para cada transformada existe uma correspondente inversa utilizada para decodificar os macroblocos tanto no codificador (para utilizar os dados posteriormente como referência temporal) quanto no decodificador (UNISINOS, 2008a).

### 3.8 Quantização

A maior parte da distorção é resultante do processo de quantização e pode ser controlada pelo valor do Parâmetro de Quantização.

O processo de quantização consiste basicamente em dividir os coeficientes transformados diminuindo sua energia. Assim, valores menos importantes para a reconstrução da imagem são zerados ou se aproximam de zero. Isto diminui a quantidade de dados necessários a serem transmitidos no fluxo MPEG-4 (UNISINOS, 2008a).

Como consequência desta diminuição de dados, há a perda de algumas informações da imagem, diminuindo sua qualidade na reconstrução. Esta relação entre tamanho e qualidade é dada pela variável denominada Parâmetro de Quantização (*Quantization Parameter*). Quanto maior o parâmetro de quantização, maior a perda gerada e conseqüentemente, menor o tamanho do fluxo gerado (Manoel, 2007).

O padrão H.264 prevê 3 diferentes quantizações. Uma que age sobre os coeficientes AC do macrobloco em quadros I ou sobre o macrobloco completo em quadros P ou B. As outras 2 trabalham com os coeficientes DC de macroblocos intra (para luma e croma) (UNISINOS, 2008a; Manoel, 2007).

Cada processo de quantização tem uma correspondente inversa utilizada para decodificar os macroblocos no decodificador.

### 3.9 Filtro Deblocking

Esta técnica é utilizada para a suavização de bordas (realizada antes de usar frames como referência). Este bloco tem por finalidade melhorar a qualidade da

imagem do bloco decodificado, aplicando uma técnica de *anti-aliasing*. O filtro de *deblocking* suaviza as bordas dos blocos realizando uma média entre seus blocos vizinhos buscando por melhorar a aparência das imagens decodificadas.

No *deblocking filter*, um macrobloco (16x16) é dividido em 4 arestas verticais e 4 arestas horizontais para *luma*, e 2 arestas para as amostras de croma. Como resultado, se obtém blocos de 4x4 para *luma* e a filtragem é feita entre estes blocos. A filtragem consiste em fazer médias entre as amostras para eliminar as mudanças bruscas no valor das amostras entre um bloco e outro devido à utilização de vetores de movimentos ou o uso de valores muito altos no processo de quantização.

### 3.10 Reordenação e Codificação de Entropia

De forma a utilizar o menor número possível de *bits* na codificação final, os coeficientes quantizados são primeiramente reordenados em uma lista de 16 elementos através da varredura em zig-zag. Os primeiros elementos da lista são os que geralmente têm valores maiores e os últimos aqueles que provavelmente têm valor zero (Manoel, 2007).

O processo de codificação por entropia consiste em aplicar os métodos de codificação CABAC e CAVLC (UNISINOS, 2008a):

- CABAC (*Context Adaptive Binary Arithmetic Coding*, Codificação Aritmética Binária Adaptável ao Contexto) é um sistema de codificação de entropia sem perdas, eliminando redundâncias. Seu processo de codificação gera dados altamente comprimidos, entretanto possui alto custo computacional.
- CAVLC (*Context-adaptive Variable-Length Coding*, Codificação de Comprimento Variável Adaptável ao Contexto). É uma técnica de compressão inerentemente sem perdas. Este método é utilizado para codificar blocos residuais que foram submetidos ao método zig-zag scan. O CAVLC é aplicado em blocos 4x4 para maior eficiência e para tentar tirar proveito das características de um bloco 4x4 quantizado, como a grande quantidade de zeros em seqüência e de pequenos valores (seqüências de 1 e -1) no final da sequência resultante do zig-zag scan. O CAVLC codifica um bloco

observando o número de coeficientes válidos (não zero) e o número de vezes que o número 1 (tanto positivo quanto negativo) aparece em seqüência e a quantidade de zeros do bloco.

### 3.11 Controle de Taxa no H.264/AVC

Em comparação com os padrões anteriores, o padrão H.264/AVC alcançou até 50% de melhora de eficiência no *bit rate*. Portanto, sua implementação é um tópico de pesquisa muito popular nos dias de hoje (Yasakethu *et al*, 2008).

O algoritmo de controle de taxa usado no H.264/AVC ajusta dinamicamente os parâmetros do codificador para alcançar o *bit rate* desejado. Ele aloca a provisão de *bit* para cada grupo de imagens, imagem individual e subimagem numa sequencia de vídeo. O controle de taxa não é uma parte do padrão H.264/AVC, mas o grupo de padrões tem publicado orientações não normativas a fim de contribuir com a implementação. Blocos baseados em esquemas de codificação de vídeo híbrido como o H.264 e MPEG-4 são considerados processos com perdas. Eles tentam alcançar a compressão não somente através da remoção da redundância temporal e informação espacial da sequencia de vídeo, mas também tenta conseguir compromisso de qualidade por meio de caminhos que teriam efeitos mínimos sobre a qualidade percebida.

Em particular o parâmetro de quantização regula o montante de detalhe espacial retido. Por exemplo, quando o parâmetro de quantização é pequeno, muito da informação é salva e quando ele é grande muito da informação é descartado para reduzir o *bit rate* pagando o preço de aumentar a distorção. Portanto, a decisão sobre qual *trade off* entre a qualidade de vídeo da sequencia de vídeo codificada e seu *bit rate codificado*, desempenha um papel significativo quando se trata de aplicações de vídeo (Yasakethu *et al*, 2008).

### 3.12 Organização dos Dados

A organização dos dados no padrão H.264 traz muita flexibilidade em relação aos padrões anteriores. Todas as estruturas de sintaxe são encapsuladas em “pacotes” de dados denominados NAL (*Network Abstraction Layer*), que podem então ser utilizados diretamente em sistemas de transporte ou armazenamento (Sullivan e Wiegand, 2005).

As unidades NAL que carregam os elementos de sintaxe da codificação de vídeo denominam-se VCL (*Camada de Codificação de Vídeo - Video Coding Layer*), enquanto as unidades que carregam informações “extras” ou conjuntos de parâmetros mais gerais (por exemplo, dados como o formato do vídeo ou o tipo de codificação de entropia) denominam-se não-VCL. As unidades não-VCL que carregam conjuntos de parâmetros têm grande importância, pois afetam um grande número de unidades NAL, além de que, essa separação por tipo de dados (VCL e Não-VCL) permite que as unidades NAL possam ser transportadas por um canal mais robusto, com menor probabilidade de perda de dados (Manoel, 2007).

A VCL organiza o vídeo hierarquicamente em objetos encapsulados, como ilustrado pela Figura 10. O MB, de tamanho fixo 16x16, é a unidade básica de codificação da VCL – ele pode ser subdividido em blocos menores para fins de predição ou transformação – mas a codificação é feita MB a MB (Manoel, 2007).

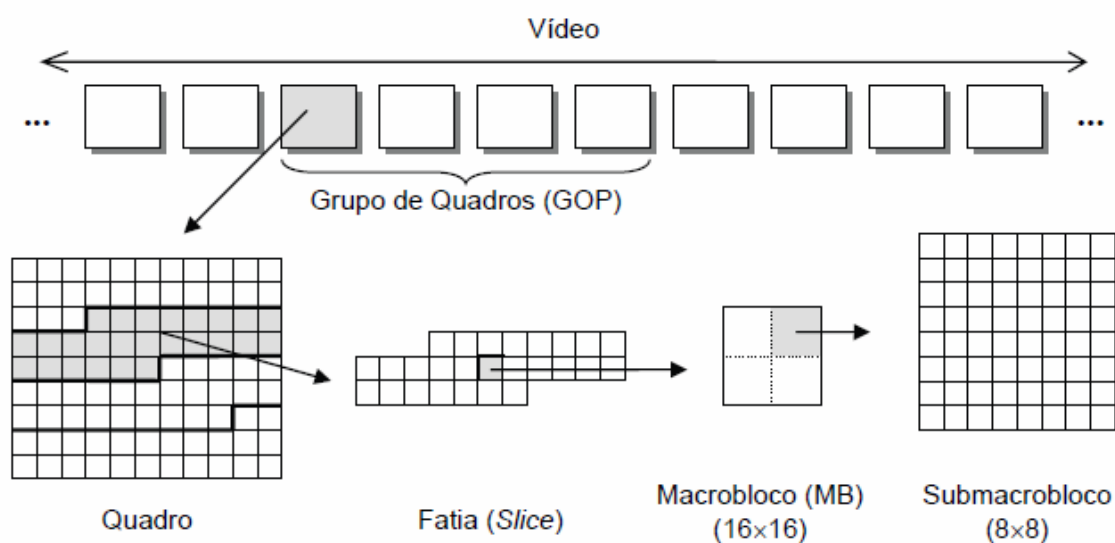


Figura 10 – Organização hierárquica da camada de codificação de vídeo.

## 4 METAHEURÍSTICAS

O termo metaheurística foi conhecido no mesmo artigo que introduziu o termo Busca Tabu (Glover, 1986) e tem sido amplamente aplicado na literatura (Glover e Laguna, 1997).

Pode-se dizer que uma metaheurística é uma estratégia-mestre que guia e modifica outras heurísticas para produzir soluções que vão além daquelas soluções normalmente geradas dentro da busca do ótimo local. As heurísticas guiadas por tal meta-estratégia pode ser de procedimento de alto nível ou pode não incluir nada mais que a descrição de passos disponíveis para transformar uma solução dentro de outra, junto com uma regra de avaliação associada (Glover e Laguna, 1997).

Metaheurísticas solucionam instâncias de problemas que se acredita serem difíceis, explorando geralmente um grande espaço de busca de soluções dessas instâncias. Esses algoritmos alcançam isso, reduzindo o tamanho efetivo do espaço a fim de explorá-lo eficientemente.

A complexidade de um problema é equivalente a complexidade do melhor algoritmo que soluciona tal problema. Um problema é tratável (ou fácil) se existir um algoritmo em tempo polinomial que o resolva. Um problema é intratável (ou difícil) quando não existir um algoritmo em tempo polinomial que o resolva.

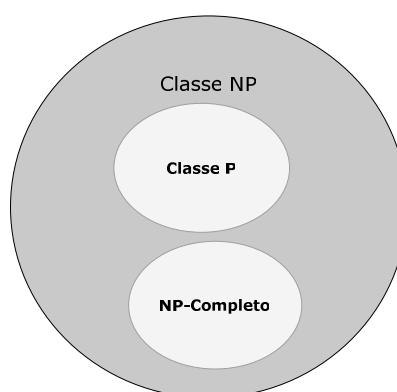


Figura 11 – Classes de complexidade de problemas de decisão.

Um importante aspecto da teoria computacional é categorizar problemas dentro de classes de complexidade. Uma classe de complexidade representa um conjunto de todos os problemas que podem ser resolvidos usando uma dada

quantidade de recursos computacionais. Existem duas importantes classes de problemas: P e NP (Garey et al, 1974).

A classe P representa uma família de problemas onde é conhecido um algoritmo em tempo polinomial que resolve esses problemas. Os problemas que pertencem a classe P são então relativamente fáceis de serem resolvidos. Exemplos: *minimum spanning tree*, *shortest path problems*, *maximum flow network*, etc.

A classe de complexidade NP (Garey et al, 1974) representa o conjunto de todos os problemas de decisão que podem ser resolvidos por um algoritmo não-determinístico em tempo polinomial. Exemplo: *0-1 knapsack problem*, problema do caixeiro viajante, problema do ciclo hamiltoniano, etc.

A questão se  $P = NP$  é um dos mais importantes questionamentos em aberto devido ao grande impacto que a resposta teria sobre a teoria da complexidade computacional.

Um problema de decisão **A ∈ NP – completo** se todos os outros problemas da classe NP (Garey et al, 1974) são reduzidos polinomialmente ao problema **A**. A Figura 11 mostra o relacionamento entre os problemas da classe P, NP e NP-completo. Se existir um algoritmo determinístico polinomial que resolva um problema NP-completo, então todos os problemas da classe NP podem ser resolvidos em tempo polinomial.

Problemas *NP-hard* são problemas de otimização que estão associados a problemas de decisão do tipo NP-completo. A maioria dos problemas de otimização do mundo real são *NP-hard* para os quais comprovadamente não existem algoritmos eficientes. Eles requerem tempo exponencial para serem resolvidos na otimalidade. Exemplos: *flow-shop scheduling*, *job-shop scheduling*, *open-shop scheduling*, *GAP (Generalized Assignment Problem)*, *QAP (Quadratic Assignment Problem)*. Metaheurísticas constituem uma alternativa importante para resolver esta classe de problemas.

A Tabela 2 mostra como o tempo de busca de um algoritmo cresce em relação ao tamanho do problema usando complexidades de tempos diferentes de um algoritmo de otimização.

Tabela 2 – Tempo de busca em função do tamanho do problema

Complexidade	Tamanho=10	Tamanho=20	Tamanho=30	Tamanho=40	Tamanho=50
$O(x)$	0,00001 s	0,00002 s	0,00003 s	0,00004 s	0,00005 s
$O(x^2)$	0,0001 s	0,0004 s	0,0009 s	0,0016 s	0,0025 s
$O(x^5)$	0,1 s	0,32 s	24,3 s	1,7 min	5,2 min
$O(2^x)$	0,001 s	1,0 s	17,9 min	12,7 dias	35,7 anos
$O(3^x)$	0,059 s	58,0 min	6,5 anos	3855 séculos	2x108 séculos

Fonte: (Talbi, 2009).

Em fim, a vasta quantidade de difíceis problemas de otimização encontrados em cenários práticos tais como, telecomunicações, logística, planejamento financeiro, transportes e produção, têm motivado o desenvolvimento de técnicas poderosas de otimização que neste caso são as metaheurísticas. Essas técnicas são normalmente o resultado de adaptação de ideias provenientes de uma variedade de áreas de pesquisa. As ideias que fundamentam a estrutura particular de cada metodologia algumas vezes vêm de conexões inesperadas. Programação de fluxo de rede, por exemplo, compartilha uma herança com modelos explorando ideias a partir da eletricidade e da hidráulica. A metaheurística denominada Arrefecimento Simulado (*simulated annealing*) é baseada em processos físicos da metalurgia. Algoritmo Genético procura imitar o fenômeno biológico da reprodução evolutiva, enquanto que a metaheurística Colônia de Formigas simula uma colônia de formigas que cooperam em uma atividade de solução para um problema em comum. A filosofia da Busca Tabu é produzir e explorar uma coleção de princípios de solução de problemas inteligentes. Neste consenso, pode-se dizer que a Busca Tabu é baseada em conceitos selecionados que une os campos da inteligência artificial e da otimização (Glover e Laguna, 1997).

## 5 BUSCA TABU

Busca Tabu é uma metaheurística que guia um procedimento de busca de heurística local para explorar o espaço de soluções que vai além da otimalidade local. O procedimento local é uma busca que usa uma operação chamada “passo” para definir a vizinhança de qualquer solução dada. Um dos principais componentes da BT é o seu uso de memória adaptativa, o qual cria um comportamento de busca



mais flexível. Estratégias baseadas em memória são portanto a marca registrada da BT (Glover e Laguna, 1997).

A palavra tabu é definida em alguns dicionários como sendo “designado como encarregado de poder sobrenatural perigoso e proibido para uso profano” ou “proibido por razões de moralidade ou por constituir um risco”. Busca Tabu dificilmente envolve referência ao sobrenatural ou considerações morais, mas ao invés disso está preocupada com imposições de restrições para guiar um processo de busca a fim de percorrer regiões difíceis. Essas restrições operam sob diversas formas, não só por exclusão direta de alternativas de busca classificadas como “proibidas”, como também por transformação de avaliações alteradas e probabilidade de seleção. Restrições são impostas ou geradas para fazer referência a estruturas de memória que são designadas para este propósito específico (Glover e Laguna, 1997).

Muitos pesquisadores têm feito contribuições para avançar no campo de aplicações da busca tabu. A Tabela 3 identifica algumas dessas contribuições e indica aplicações específicas beneficiadas a partir delas (Glover *et al*, 1995).

Tabela 3 – Algumas aplicações da Busca Tabu

Área	Descrição	Referência
Programação / agendamento de tarefas	Agendamento de Escala de funcionários; Programação de <i>flow shop</i>	Glover e McMillan (1986). Wildmer e Hertz (1989 ). Taillard (1990).
	Programação de job shop com restrições de ferramenta;	Wildmer (1991).
	Programação de comboios;	Bove et al (in press.).
	Programação de única máquina;	Laguna et al (1991).
	Programação Just-in-time	Laguna e Gonzalez-Velard (1991).
	Problema do tempo de fluxo de múltiplas máquinas ponderadas;	Barnes e Laguna (1993).
	Programação de job shop de recurso flexível;	Daniels and Mazzola (1993).
	Programação de job shop;	DelrAmico e Trubian (1993). Muller e Gómez (2006).
	Programação de única máquina (análise objetiva)	Laguna e Glover (1993).
	Programação de recursos;	Mooney e Rardin (1993).

Área	Descrição	Referência
	Trabalhos seqüenciais com prazo e tempo de setup	Woodruff e Spearman (1992).
Transporte	Problema do caixeiro viajante	Malek et al (1989); Glover (1992).
	Problema do roteamento de veículo	Gendreau et al (in press); Gendreau et al (2008); Osman (1993). Semet e Taillard (1993).
Projeto de circuitos	Problema de atribuição quadrática	Skorin-Kapov (1990). Taillard (1991). Chakrapani e Skorin-Kapov (1993).
	Projeto de circuito eletrônico	Bland e Dawson (1991).
Telecomunicações	Atribuição de caminho	Oliveira e Stroud (1989). Anderson et al (1993).
	Empacotamento de banda	Glover e Laguna (1993).
Gráficos e mapas	Clusterização	Glover et al (1985). Hansen et al (in press).
	Coloração de mapas	Hertz e de Werra (1987). Hertz et al (1994)
	Conjunto estável em grandes mapas (Stable set in large graphs)	Friden et al (1989).
	Problema do clique máximo	Gendreau et al (1993).
Lógica probabilística e sistemas especialistas	Problema satisfatibilidade máxima	Hansen and Jaunard (1990).
	Lógica Probabilística	Jaumard et al (1991).
	Lógica probabilística / sistemas especialistas	Hansen et al (1992).
Redes Neurais	Aprendizagem em memória associativa	De Werra e Hertz (1989).
	Problema de otimização não convexo	Beyer e Ogier (1991).
Outros	Problema da mochila	Dammeyer e Voss (1993).
	Arredondamento controlado em larga escala ( <i>large-scale controlled rounding</i> )	Kelly et al (1993).
	Problema generalizado de taxa fixa ( <i>Generalized fixed charge problem</i> )	Sun and McKeown (1993).

Fonte: (Glover et al, 1995).

A BT busca uma solução melhor a partir de uma solução inicial informada. A solução inicial não precisa ser boa, mas deve ser uma solução viável e pode ser construída usando-se uma heurística ou de maneira aleatória. Caso a BT seja usada em um modelo onde já exista uma aplicação prática, pode-se utilizar uma solução já existente como solução inicial (Glover *et al*, 1995).

A BT parte do pressuposto que uma vizinhança pode ser construída para identificar soluções adjacentes que podem ser alcançadas a partir de qualquer solução corrente. Trocas ou *swaps* são frequentemente usados para definir vizinhanças em problemas de permutação, identificando movimentos que levam de uma solução para a próxima. É associado a cada *swap* um valor de mudança, o qual representa a alteração no valor da função objetivo. Os valores de mudança geralmente fornecem uma base fundamental para avaliar a qualidade da mudança, embora outro critério também possa ser importante (Glover *et al*, 1995).

A principal maneira de explorar memória na BT é classificar um subconjunto de movimentos numa vizinhança como proibidos (ou tabu). A classificação depende do histórico da busca, e particularmente sobre a frequência que certo movimento ou componente de solução, chamados atributos, têm participado da geração de soluções passadas. Lista tabu são tópicos de uma importante exceção. Quando um movimento tabu tem uma avaliação suficientemente atrativa, como por exemplo, onde ele poderia resultar numa solução melhor que qualquer outra já encontrada, então a lista tabu pode ser substituída. Uma solução que permite que tal substituição ocorra é chamada de “critério de aspiração” (Glover *et al*, 1995).

A estratégia da BT em instituir penalidades apenas sob condições particulares é usada para preservar a agressividade da busca.

Na prática, obviamente, a busca tem que terminar em algum ponto. Os critérios de parada mais comumente utilizados na BT segundo Glover e Kochenberger (2003) são:

- Após um número fixo de iterações ou uma quantidade fixa de tempo de CPU;
- Após um certo número de iterações sem melhoria (*nbmax*) no valor da função-objetivo (critério usado na maioria das implementações);

- Quando o objetivo alcança um valor limiar pré-estabelecido.

Em esquemas complexos de BT, geralmente a busca termina após completar uma sequência de fases sendo a duração de cada fase determinada por um dos critérios acima (Glover e Kochenberger, 2003).

Na Tabela 4 podemos ver um exemplo de algoritmo da Busca Tabu.

Tabela 4 – Algoritmo clássico da Busca Tabu

---

Busca Tabu
<p><math>s = s_0</math> ; /* Solução inicial */</p> <p>Inicie Lista Tabu, memória de médio termo e memória de longo ;</p> <p><b>Repeat</b></p> <p style="padding-left: 20px;"><b>Find</b> melhor vizinho admissível <math>s'</math> ; /* não-tabu ou possui critério de aspiração */</p> <p style="padding-left: 20px;"><math>s = s'</math>;</p> <p style="padding-left: 20px;"><b>Update</b> Lista Tabu, condições de aspiração, memória de médio e longo termo;</p> <p style="padding-left: 20px;"><b>If</b> possui critério de intensificação <b>Then</b> intensificação ;</p> <p style="padding-left: 20px;"><b>If</b> possui critério de diversificação <b>Then</b> diversificação ;</p> <p><b>Until</b> critério de parada satisfeito;</p> <p><b>Output</b> melhor solução encontrada;</p>

---

Fonte: (Talbi, 2009)

Segundo Talbi (2009), as principais implementações que uma Busca Tabu deve ter são mostradas na Tabela 4 que são:

- Lista Tabu (*short-term memorie*): O objetivo do uso de memória de curto prazo (*short-term memorie*) é prevenir a busca de visitar novamente soluções previamente visitadas. Armazenar a lista de todas as soluções visitadas não é prático por questões de eficiência.
- Critério de Aspiração (*Aspiration Criterion*): Um critério de aspiração comumente usado consiste em selecionar um movimento tabu caso ele gere uma solução melhor que a melhor solução encontrada. Outro critério de aspiração pode ser um movimento tabu que produz uma melhor solução entre um conjunto de soluções de processamento de um dado atributo.

Talbi (2009) fala que alguns mecanismos avançados são geralmente introduzidos na Busca Tabu para lidar com a intensificação e a diversificação da busca:

- Intensificação (medium-term memory): Armazena as soluções de elite (as melhores) encontradas durante a busca. A ideia é priorizar os atributos do conjunto de soluções de elite, geralmente na forma de probabilidade ponderada. A busca é influenciada por esses atributos.
- Diversificação (*long-term memorie*): Armazena informações sobre as soluções visitadas ao longo da busca. Explora áreas não visitadas do espaço de soluções. Por exemplo, ele irá desestimular os atributos das soluções de elite nas soluções geradas para diversificar a busca para outras áreas do espaço de busca.

## 6 ALGORITMO GENÉTICO

O Algoritmo Genético (AG) refere-se ao modelo introduzido e estudado por John Holland (Holland 1975) que foi desenvolvido por ele e pelos seus alunos e colegas. Deste estudo resultou o livro “*Adaptation in Natural and Artificial systems*”.

Os AGs são uma classe muito popular dos Algoritmos Evolucionários, que são baseados na noção de competição. São os algoritmos mais estudados baseados em população. O seu sucesso em resolver problemas difíceis de otimização em vários domínios promoveu o campo conhecido como computação evolucionária (Talbi, 2009).

Os algoritmos evolucionários são baseados na evolução de uma população de indivíduos. Inicialmente esta população é usualmente gerada de forma aleatória. Cada indivíduo da população é a versão codificada de uma tentativa de solução. O AG possui uma Função Objetivo associada a um valor de aptidão (*fitness*) sendo que cada indivíduo representa uma solução do problema. A cada passo, indivíduos são selecionados para formar os pais, seguindo o paradigma de seleção o qual indivíduos com melhor aptidão são selecionados com uma maior probabilidade. Então, indivíduos selecionados são reproduzidos usando operadores de variação, que são o *crossover* e a mutação, para gerar novos indivíduos ou filhos. Finalmente uma substituição é aplicada para determinar quais indivíduos da população sobreviverão a partir dos filhos e dos pais. Esta iteração representa o que

chamamos de geração. Este processo é iterado até que se alcance um critério de parada (Talbi, 2009). A Tabela 5 ilustra um *template* do algoritmo evolucionário.

Tabela 5 – *Template* de um algoritmo Evolucionário

Algoritmo Evolucionário	
Gerar (P(0)); /* população Inicial	
t = 0;	
<b>While not</b> critério de fim (P(t)) <b>Do</b>	
Avaliar (P(t));	
P'(t)	= Seleção (P(t));
P'(t)	= Reprodução (P(t)); Avaliação (P'(t));
P(t + 1)	= Substituição (P(t), P'(t) );
t = t + 1;	
<b>End While</b>	
<b>Output</b> melhor indivíduo ou melhor população encontrada.	

Fonte: (Talbi,2009).

## 6.1 Codificação

O AG é uma técnica de pesquisa utilizada na computação para encontrar soluções aproximadas para problemas complexos não lineares. O AG é inspirado no processo *Darwiniano* de seleção natural e utiliza conceitos desta teoria onde as soluções do problema são comparadas a indivíduos em um ambiente competitivo. Os AGs não necessitam de muito conhecimento sobre o problema a ser otimizado e não lidam diretamente com os parâmetros do problema. Ele trabalha com códigos que representam os parâmetros.

Durante os últimos anos, tem-se observado uma expansão na utilização de Algoritmos Evolutivos (Deb, 2001), especialmente os Algoritmos Genéticos, em problemas de otimização. Uma das características mais importantes dos AGs é que possibilitam encontrar soluções ótimas ou adequadas para um problema sem usar cálculos mais complexos como uso de derivadas de funções (Goldberg, 1989).

O ponto de partida para a aplicação de Algoritmos Genéticos a um problema de explosão combinatória é a representação do problema a ser analisado, de maneira que os algoritmos genéticos possam atuar adequadamente sobre ele.

Os Algoritmos Genéticos processam populações de indivíduos chamados cromossomos. O cromossomo é uma estrutura de dados, geralmente vetores ou cadeia de valores binários, números reais ou combinação de ambas, que representa uma possível solução do problema. Em geral, o cromossomo representa o conjunto de parâmetros da função objetivo a ser otimizada (que pode ser maximizada ou minimizada). O conjunto de todas as configurações que o cromossomo pode assumir forma o seu espaço de busca. Se o cromossomo representa  $n$  parâmetros de uma função, então o espaço de busca é um espaço com  $n$  dimensões.

Um algoritmo de otimização global deve ser capaz de explorar pontos inteiramente novos dentro do espaço de busca, bem como intensificar a busca em determinadas regiões consideradas promissoras. Esse mecanismo de diversificação e intensificação é obtido nos algoritmos genéticos pela correta aplicação dos operadores genéticos.

## 6.2 Inicialização

O Algoritmo Genético começa com uma população inicial de  $n$  indivíduos. Esses indivíduos podem ser gerados de forma aleatória, dentro dos limites do problema, quando não existir conhecimento prévio sobre a região do espaço de pesquisa onde se encontra a solução do problema. No entanto, se houver um conhecimento *a priori* sobre a região em que está localizada a solução e forem conhecidas soluções aceitáveis que podem estar próximas da solução ótima, os indivíduos iniciais podem ser definidos de forma determinística.

## 6.3 Avaliação

Algoritmos genéticos necessitam do valor de uma função objetivo para cada membro da população, que deve ser um valor não negativo. Nos casos mais simples, usa-se justamente o valor da função que se quer otimizar. A Função Objetivo calcula para cada indivíduo, uma medida de quão bem adaptado ao ambiente ele está, ou seja, suas chances de sobreviver no ambiente e reproduzir-se, passando parte do seu material genético a gerações posteriores.

A avaliação de cada indivíduo resulta num valor denominado aptidão ou *fitness*.

## 6.4 Operador de Seleção

O operador de seleção é utilizado para melhorar a qualidade média da população, dando a indivíduos com melhores *fitness*, uma maior probabilidade de ser levado para uma próxima geração de indivíduos. A seleção incide na procura de soluções promissoras dentro do espaço de busca.

### 6.4.1 Seleção por Roleta

É o método de seleção original proposto por *Holland* (1975) para os algoritmos genéticos. A probabilidade de um indivíduo ser selecionado é proporcional ao valor do seu *fitness*.

Na seleção por roleta, indivíduos excepcionais irão introduzir uma tendência no início da pesquisa que pode causar uma convergência prematura e uma perda de diversidade. Além disso, quando todos os indivíduos são igualmente ajustados, essa estratégia de seleção não introduz uma pressão suficiente para selecionar os melhores indivíduos (Talbi, 2009).

### 6.4.2 Seleção por Torneio

Neste método  $n$  indivíduos da população são escolhidos aleatoriamente e com a mesma probabilidade. O indivíduo com maior aptidão dentre estes  $n$  indivíduos é selecionado para a população intermediária. O processo se repete até que a população intermediária seja preenchida.



### 6.4.3 Seleção por Amostragem Estocástica

Este método é uma variação do Método da Roleta, porém em vez de uma única agulha,  $n$  agulhas igualmente espaçadas são utilizadas, onde  $n$  é o número de indivíduos a serem selecionados. Assim, em vez de  $n$  vezes, a roleta é girada uma única vez.

## 6.5 Operador de Cruzamento

O operador de cruzamento ou *crossover* consiste na troca de material genético entre os pais selecionados com o objetivo de se formarem descendentes com melhor *fitness*, empregando o conceito de evolução das espécies. Este operador gera novas soluções (filhas) a partir de soluções escolhidas da lista de soluções (pais). O operador de cruzamento possui diferentes variações, muitas delas específicas a um determinado problema.

Alguns pontos importantes devem ser levados em consideração quanto ao uso do operador de *crossover*:

- a) Herança: o operador de *crossover* deve receber como herança um material genético de ambos os pais.
- b) Validade: O operador de *crossover* deve produzir soluções válidas. Isso nem sempre é possível para problemas de otimização com restrições.

É importante notar que o cruzamento possui uma probabilidade associada, geralmente um valor próximo de 1.

### 6.5.1 Cruzamento de Um Ponto

A forma mais básica de cruzamento é conhecida como cruzamento em um ponto ou *1-point crossover* e sua forma generalizada é o *n-point crossover*. O *1-point crossover* consiste em:

- a) Escolher arbitrariamente dois indivíduos na lista de soluções;

- b) Escolher dentro da cadeia do indivíduo uma posição  $k$  chamada posição de cruzamento;
- c) Criar novos descendentes trocando as cadeias parciais de cada um dos indivíduos.

A Tabela 6 mostra um exemplo de *crossover* de um ponto.

Cromossomo 1	10010   00100110110
Cromossomo 2	11011   11000011110
Filho 1	10010   11000011110
Filho 2	11011   00100110110

### 6.5.2 Cruzamento Aritmético

Este tipo de cruzamento tenta calcular a média dos elementos dos pais. Dado dois pais  $p_1$  e  $p_2$ , o cruzamento aritmético pode gerar até dois filhos  $o_x$  e  $o_y$  usando a média ponderada (Talbi, 2009):

(5)

$$o_x = \alpha x_{1i} + (1 - \alpha)x_{2i}$$

$$o_y = (1 - \alpha)x_{1i} + \alpha x_{2i}$$

(6)

Onde  $\alpha \in [0,1]$ ,  $x_{1i}$  é o valor do alelo  $i$  do primeiro pai,  $x_{2i}$  é o valor do alelo  $i$  do segundo pai.

## 6.6 Operador de Mutação

O operador de mutação pode ser entendido tanto como um diversificador como um intensificador de busca. A mutação diversifica quando introduz uma

informação inteiramente nova no indivíduo e, conseqüentemente à população. Por outro lado, quando apenas aplica um ruído à solução contida no indivíduo, a mutação é um mecanismo intensificador de busca na vizinhança dessa solução.

No processo de mutação, basicamente, seleciona-se uma posição num cromossomo e muda-se o valor do gene correspondente aleatoriamente para um outro possível. O processo é geralmente controlado por um parâmetro fixo que indica a probabilidade de um gene sofrer mutação.

### 6.6.1 Mutação Uniforme

A mutação uniforme é geralmente usada quando um cromossomo é codificado na forma de números reais.

Nesse tipo de mutação sorteia-se qual dos cromossomos sofrerá mutação e em seguida sorteia-se um valor para esse cromossomo dentro do seu espaço de definição, neste caso temos  $\mathbf{x}' = x_1, \dots, x_k', \dots, x_n$  sendo  $x'$  um valor aleatório escolhido dentro do seu espaço de definição.

## 6.7 Operador de Elitismo

O operador de elitismo mantém as melhores soluções encontradas previamente nas gerações seguintes. Isto implica que a melhor solução não se deteriora nas gerações seguintes.

## 6.8 Parâmetros Genéticos

Os principais parâmetros genéticos são:

- Tamanho da população: O tamanho da população afeta o desempenho global e a eficiência dos algoritmos genéticos. Com uma população pequena o desempenho pode cair, pois deste modo a população fornece uma pequena cobertura do espaço de busca do problema. Uma grande população

geralmente fornece uma cobertura representativa do domínio do problema, além de prevenir convergências prematuras para soluções locais ao invés de globais.

- Taxa de cruzamento: Quanto maior for a taxa de cruzamento, mais rapidamente novas estruturas serão introduzidas na população. Mas se esta for muito alta, estruturas com boas aptidões poderão ser retiradas mais rapidamente, a maior parte da população será substituída, podendo ocorrer perda de estruturas de alta aptidão. Com um valor baixo, o algoritmo pode tornar-se muito lento ou estagnar.
- Taxa de mutação: Uma baixa taxa de mutação pode evitar que a pesquisa fique estagnada, mas com uma taxa muito alta a pesquisa torna-se essencialmente aleatória.
- Taxa de substituição: Controla a porcentagem da população que será substituída durante a próxima geração.
- Condição de parada: normalmente usa-se o critério do número máximo de gerações ou um tempo limite de processamento como sendo a condição de parada de um algoritmo genético.

## **7 SMC – SIMULADOR DE METAHEURÍSTICAS APLICADO A UM CODEC**

Algoritmos para compressão e descompressão de vídeo, os chamados CODECs têm sido continuamente melhorados ao longo da última década para atender as demandas do mercado (Golston e Rao, 2006). Um dos últimos CODECs desenvolvido foi o chamado de padrão H.264/MPEG-4 AVC (ITU-T, 2007), o qual foi escolhido como padrão do Sistema Brasileiro de TV Digital (SBTVD) (ABNT NBR 15601:2008; ABNT NBR 15602-1:2008).

O padrão H.264 permite que vários de seus parâmetros possam ser configurados o que o torna altamente flexível no sentido de obter uma boa qualidade e compressão de vídeo e conseqüentemente altera seu desempenho. Na norma da ITU-T (2007) são definidos todos os parâmetros configuráveis do H.264.

Dentro deste contexto, vários autores tem proposto algoritmos para otimizar a performance do CODEC de vídeo H.264 (Yasakethu *et al*, 2008; Cermak *et al*, 2011; Huang *et al*, 2006; Nemethova *et al*, 2004; Correia e Perreira, 2003). Para avaliar o desempenho e a qualidade da imagem produzida por esses algoritmos são usadas métricas objetivas e subjetivas (Sikora, 2005; Zhang *et al*, 2003; Moriyoshi *et al*, 2000; Malvar *et al*, 2003; ITU-R, 2002).

Este trabalho apresenta um algoritmo híbrido composto por duas metaheurísticas, Busca Tabu e Algoritmo Genético, denominado Simulador de Metaheurísticas aplicado a um CODEC (SMC) cujo principal objetivo é encontrar a melhor configuração de parâmetros do H.264 a fim de ajudar a melhorar seu desempenho e a qualidade da imagem codificada. São considerados neste estudo seis parâmetros de configuração diretamente relacionados ao codificador H.264 que é o CODEC da TV digital Brasileira desenvolvido pelo projeto DigConv (Unisinos, 2008), que são: *bit rate* (BR), *frame rate* (FR), parâmetros de quantização de *frames* tipo I (QI), *frames* tipo P (QP) e *frames* tipo B (QB) e número de *frames* tipo B (PF) dentro de um GOP (*Group of Picture*).

Para isso foi feito um estudo da dinâmica de comportamento entre os parâmetros estudados a fim de descobrir um padrão de comportamento entre eles.

## 7.1 Arquitetura Sistêmica do SMC

A arquitetura sistêmica do SMC é mostrada na Figura 12. Essa é a arquitetura do CODEC de vídeo H.264 da TV Digital Brasileira desenvolvido pelo projeto DigConv (Unisinos, 2008). Sobre esses módulos é que os parâmetros otimizados pelo SMC atuarão. Note que o Decodificador de Vídeo que faz parte deste CODEC não é tratado nesse modelo.

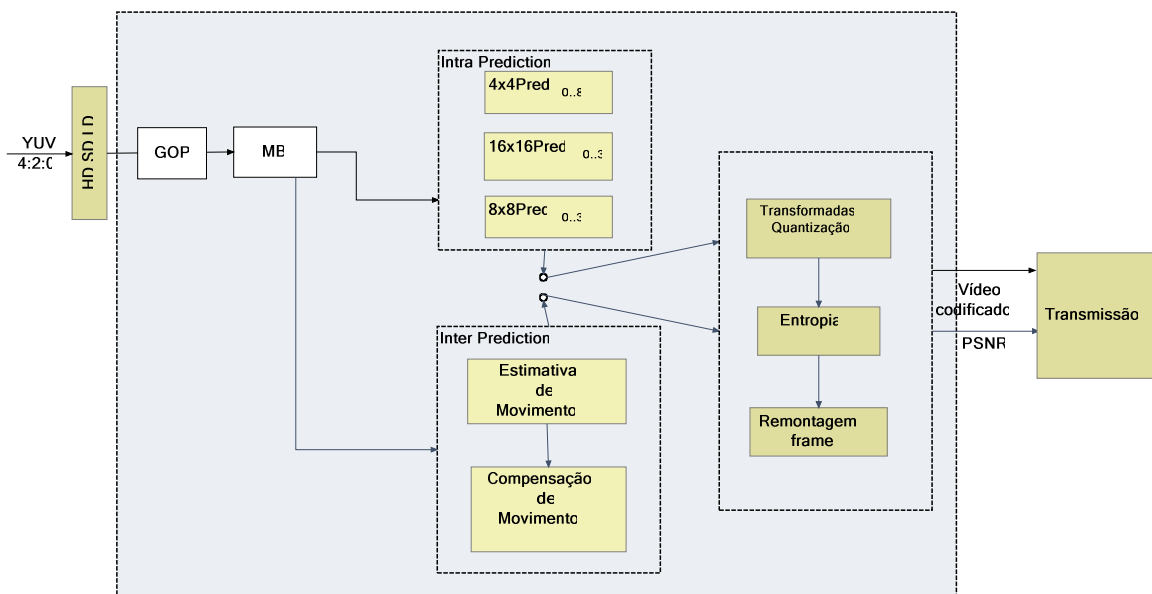


Figura 12 – Arquitetura Sistêmica do SMC.

A seguir serão detalhados cada um dos parâmetros otimizados pelo SMC e como eles vão atuar ou influenciar em cada um dos módulos que fazem parte do CODEC H.264 do SBTVD mostrado na Figura 12. Esses módulos já foram detalhados na seção 3 que fala sobre o padrão H.264, agora eles serão abordados dentro da descrição de cada parâmetro de configuração tratado pelo SMC a fim de que tenhamos uma ideia de como os parâmetros de codificação vão atuar sobre esses módulos.

#### 7.1.1 Parâmetro de configuração de *frames* tipo B

O parâmetro PF configura a quantidade consecutiva de *frames* tipo B (Bi-predito) que devem fazer parte de um GOP (Group Of Picture), conforme mostrado na Figura 13. Um aumento no número de *frames* B causa uma degradação na qualidade da imagem. O CODEC H.264 do projeto DigConv permite configurar o tamanho de um GOP e dentro deste, configura-se o parâmetro PF.

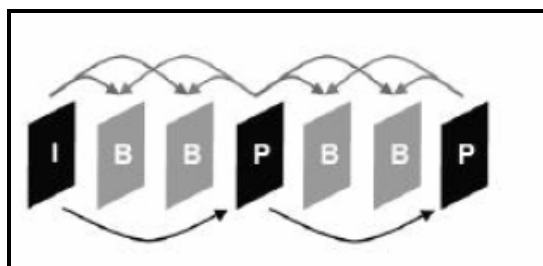


Figura 13 – GOP - Group Of Picture (Golston and Rao, 2006).

No exemplo mostrado na Figura 13 temos um GOP de tamanho 7, sendo o parâmetro PF igual a 2, ou seja, dentro de um grupo de 7 frames, temos 2 frames B consecutivos seguidos de um frame P. Um GOP sempre iniciará com um frame I. O SMC porém usa um GOP de tamanho fixo igual a 10, onde o parâmetro PF pode variar de 0 a 8.

O parâmetro PF vai atuar diretamente na decomposição do vídeo de entrada em frames tipo I, P e B e estes frames por sua vez são decompostos em macroblocos.

O SMC utiliza como entrada vídeos no padrão YUV 4:2:0 que pode ser do tipo HD, SD ou LD. O módulo de Particionamento é encarregado de converter o vídeo de entrada em sua unidade básica de codificação que é o MacroBloco (MB) e cada MB consistirá de uma matriz de 16x16 amostras luma (Y') e duas matrizes de 8x8 amostras croma (uma para Cb e outra para Cr). Este módulo gerará MBs do tipo P, B e I. Após o MB ser extraído ele é codificado pelos módulos denominados *Inter Prediction* e *Intra Prediction*.

### 7.1.2 Parâmetro *Bit Rate*

O *bit rate* reflete ganho de compressão no domínio espacial e temporal. Além disso, o desempenho do codificador é dependente das características do movimento. A redução do *bit rate* provoca a perda da informação espacial e temporal que é normalmente irritante para os telespectadores. Koumaras *et al* (2005), afirma que a melhoria da qualidade de vídeo não é significativa para taxas de bits mais altas do que um limiar e que esta também é dependente da atividade espacial e temporal do vídeo. Koumaras *et al* (2005) também mostra que a variação do nível de qualidade da imagem em relação a taxa de bits é uma função não linear crescente com uma inclinação que depende da dinâmica do vídeo.

Pode-se dizer que o *bit rate* é um parâmetro de extrema importância na percepção da qualidade da imagem e está diretamente ligado aos módulos do CODEC que tratam a redundância espacial, módulo *Intra Prediction*, e redundância temporal, módulo *Inter Prediction*.

O módulo *Intra Prediction* tem como principal função eliminar a redundância espacial (descrita no item 3.1.1) e recebe como entrada MBs do tipo I, P ou B, que podem ser particionados nos formatos: 4x4, 8x8 ou 16x16. O formato 4x4 possui até nove opções de Predição Intra conforme Figura 7 e os formatos 8x8 e 16x16 possuem 4 opções de Predição Intra conforme Figura 8. Como saída, este módulo fornece o que chamamos  $MB_{intra}$ .

O módulo *Inter Prediction* tem como principal função eliminar a redundância temporal (descrita no item 3.1.2) e recebe como entrada MBs do tipo P ou B, que podem ser particionados em até 259 partições/subpartições diferentes. Neste módulo serão realizadas as etapas de Estimação de Movimento e Compensação de Movimento. Nessas duas etapas são encontrados um dos processos mais custosos em termos de processamento, que consiste em fazer uma busca na região em torno de um MB em todos os quadros de referência, para determinar o vetor de movimento que minimiza o erro residual (dada uma determinada métrica). Quanto maior a região de busca e em quanto mais quadros de referência se procurar, melhor será o resultado, porém com um custo computacional maior, o resultado é um Vetor de Movimento e um bloco residual.

A Estimação de Movimento consiste em procurar, no quadro de referência, um bloco mais semelhante possível ao bloco atual, através do processo de interpolação. Ele utiliza comparação entre blocos vizinhos para encontrar o mais parecido. Após encontrar o bloco do *frame* de referência que mais se parece com o atual, ele passa para a Compensação de Movimento criar os vetores de movimento;

Na etapa de Compensação de Movimento, após escolhido o bloco de predição, este é subtraído do bloco atual, para se obter um bloco residual ( $BlocoRes = BlocoAtual - BlocoRef$ ). É calculada a diferença de posição entre o bloco de referência e o bloco atual, o que resulta em um vetor de movimento.

### 7.1.3 Parâmetros de quantização de *frames* do tipo I, P e B

O CODEC H.264 do projeto DigConv (Unisinos, 2008) permite que sejam parametrizados individualmente os parâmetros de quantização de *frames* I, P e B, que chamaremos de QI, QP e QB respectivamente. Esses parâmetros atuam



diretamente no módulo de Transformadas e Quantização do CODEC. Contudo, é válido ressaltar que a maior parte da distorção é resultante do processo de quantização e conseqüentemente controlada pelo valor do parâmetro de quantização.

O módulo de Transformada e Quantização são processos complementares. Uma transformada matemática baseada em cossenos realiza a transformada dos *pixels* após o processo de predição (inter ou intra). Esta transformação se baseia em uma compactação dos dados que possui reversão (modo de recuperar os dados originais). As transformações utilizadas são baseadas em DCT (Transformada Discreta de Cosseno) e na transformada *Hadamard* (Richardson,2002).

Para o processo de quantização, são necessários o bloco de entrada e os valores de QI, QP e QB, que são os fatores que equilibram a relação qualidade de imagem por compressão. Na quantização é que efetivamente se troca qualidade por taxa. Quanto maior a quantização, mais bits se perdem, portanto, maior é a degradação da imagem. O padrão H.264 prevê 3 diferentes quantizações, uma que age sobre os coeficientes AC do macrobloco em quadros I ou sobre o macrobloco completo em quadros do tipo P ou B. As outras duas trabalham com os coeficientes DC de macroblocos codificados pela Intra Prediction (para amostras Luma e Croma).

O processo de codificação por entropia tem como entrada MBs quantizados provenientes do módulo de quantização e que serão codificados usando os métodos de codificação CABAC e CAVLC, já o processo de remontagem de frame constrói a *stream* MPEG-4 que será empacotada e enviada pelo meio de transmissão. Como saída desses processos temos o vídeo codificado e o PSNR.

#### 7.1.4 Parâmetro *Frame Rate*

O *frame rate* se refere a uma única imagem consecutiva chamada *frame* produzida dentro de uma unidade de tempo. O *frame rate* é mais frequentemente dado em *frames* por segundo (fps) ou alternativamente em *Hertz* (Hz). Koumara *et al* (2005) afirma que quando o *frame rate* fica abaixo de 25 fps a qualidade da imagem percebida é considerada inaceitável.

## 7.2 Formulação do Problema

Ultimamente, tem-se dado ênfase no desenvolvimento de métodos e técnicas para avaliar o nível de qualidade percebida de um conteúdo de vídeo. Esses métodos podem ser agrupados em duas principais categorias, a categoria subjetiva e a objetiva. Os métodos de teste subjetivos envolvem pessoas que assistem a vídeos e avaliam sua qualidade na forma como esses vídeos são percebidos por eles dentro de um ambiente específico e controlado. O MOS (do inglês, *Mean Opinion Score*) é considerado como o método mais confiável de medida de qualidade de vídeo e tem sido aplicado nas técnicas subjetivas mais conhecidas: Single Stimulus Continue Quality Evaluation (SSCQE) e Double Stimulus Continue Quality Evaluation (DSCQE). Contudo, o método MOS é dispendioso e demorado e por esta razão tem havido um grande esforço por parte da comunidade científica em desenvolver métodos de avaliação mais baratos e de mais fácil aplicação. Essas técnicas têm emulado com sucesso os resultados da avaliação de qualidade subjetiva. Os métodos objetivos são classificados de acordo com a disponibilidade do sinal de vídeo original que é considerado de alta qualidade (Koumaras *et al*, 2005).

Os principais métodos objetivos propostos requerem uma fonte de sequência de vídeos sem distorção como uma entidade de referência no processo de avaliação da qualidade e devido a isso são caracterizados como métodos de referência completos. Esses métodos são baseados no quadro de sensibilidade ao erro e as métricas mais amplamente utilizadas são o PSNR e MSE, contudo esse tipo de métrica tem sido seriamente criticado por não produzir uma medida confiável na qualidade de vídeo percebida e devido a isso tem-se gerado um grande esforço em se criar métodos de avaliação que emulem características do sistema visual humano (HVS, do inglês Human Visual System), mas o que tem-se notado é que esses métodos complicados de HVS não produzem resultados mais confiáveis do que medidas matemáticas simples, como por exemplo, o PSNR (Koumaras *et al*, 2005).

Ries *et al* (2007) propõe uma métrica universal baseada nos parâmetros de movimento com o objetivo de estimar a qualidade de vídeo do *streaming* de vídeo móvel a nível de usuário (qualidade percebida do serviço) para qualquer configuração possível de CODEC em rede 3G e para qualquer tipo de conteúdo.

### 7.2.1 Formulação Matemática

A equação (7) pode ser vista como uma métrica objetiva que avalia a qualidade de vídeo de um CODEC de vídeo H.264/AVC e que será posteriormente comparada ao PSNR. Quanto maior o valor da FO, maior será a qualidade do vídeo codificado.

Sejam as variáveis de decisão:

$FR_{n,f}$  = *frame rate* para o nível n e formato de vídeo f;

$BR_{n,p}$  = *bit rate* para o nível n do perfil p;

$QI$  = parâmetro de quantização de *frames* tipo I;

$QP$  = parâmetro de quantização de *frames* tipo P;

$QB$  = parâmetro de quantização de *frames* tipo B;

$PF$  = quantidade de *frames* tipo B em um GOP;

$\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6$  = pesos pré-definidos e não tendenciosos das variáveis de

decisão BR, FR, QI, QP, QB e PF respectivamente;

Sejam os índices:

$f$  = formato do vídeo, onde: 1=QCIF, 2=CIF, 3=525 SD, 4=720p HD, 5=1080 HD;

$p$  = perfil do padrão H.264 , onde: 0=baseline, 1=main, 2=high;

$n$  = identifica o nível do padrão H.264;

Função Objetivo:

$$\text{Maximizar FO} = \alpha_1 BR_{n,p} + \alpha_2 FR_{n,f} + \alpha_3 \frac{1}{QI} + \alpha_4 \frac{1}{QP} + \alpha_5 \frac{1}{QB} + \alpha_6 \frac{1}{PF}$$

(7)

Sujeito a:

$$30 \leq BR_{n,p} \leq 300000$$

(8)

$$15 \leq FR_{n,f} \leq 172$$

(9)

$$0 \leq QI \leq 51$$

(10)

$$0 \leq QP \leq 51 \tag{11}$$

$$0 \leq QB \leq 51 \tag{12}$$

$$0 \leq PF \leq 8 \tag{13}$$

$$\alpha_1 > 0 \tag{14}$$

$$\alpha_2 > 0 \tag{15}$$

$$\alpha_3 > 0 \tag{16}$$

$$\alpha_4 > 0 \quad (17)$$

$$\alpha_5 > 0 \quad (18)$$

$$\alpha_6 > 0 \quad (19)$$

A Função Objetivo dada pela equação (7) busca a maximização da qualidade de um vídeo codificado no padrão H.262. Esta função é composta por seis variáveis, onde cada uma delas representa um parâmetro de codificação essencial do CODEC H.264/AVC.

As restrição (8) representa os possíveis os valores assumidos pelo *bit rate* ao se codificar um vídeo dentro do nível (n) e perfil (p) do CODEC H.264. Conforme já falado na seção 3.3, o nível e o perfil do CODEC H.264 estabelece os valores permitidos para alguns parâmetros de codificação dentre eles o *bit rate*.

Os valores de *bit rates* permitidos pelo CODEC H.264 e que estão de acordo com seu nível e perfil são mostrados na Tabela 7.

Tabela 7 – Variação do *bit rate* de acordo com o nível e perfil do H.264.

Nível (Profile Level)	Bit rate (Kbps)	
	Baseline, Main	Perfil High
1	30 a 64	60 a 80
1b	64 a 128	80 a 160
1.1	128 a 192	160 a 240
1.2	192 a 384	240 a 480
1.3	384 a 768	480 a 960
2	768 a 2000	960 a 2500
2.1	2000 a 4000	2500 a 5000

Bit rate (Kbps)		
Nível (Profile Level)	Perfil	
	Baseline, Main	High
2.2	2000 a 4000	2500 a 5000
3	4000 a 10000	5000 a 12500
3.1	10000 a 14000	12500 a 17500
3.2	14000 a 20000	17500 a 25000
4	14000 a 20000	17500 a 25000
4.1	20000 a 50000	25000 a 62500
4.2	20000 a 50000	25000 a 62500
5	50000 a 135000	62500 a 168750
5.1	135000 a 240000	168750 a 300000

Fonte: ITU-T (2007).

As restrição (9) representa os possíveis valores assumidos pelo *frame rate* ao se codificar um vídeo no formato f e nível n do CODEC H.264. Esses valores são mostrados na Tabela 8.

A restrição (10) garante que o parâmetro de quantização de *frames* tipo I (QI), varie dentro do intervalo de números inteiros positivos [0, 51].

A restrição (11) garante que o parâmetro de quantização de *frames* tipo P (QP), varie dentro do intervalo de números inteiros positivos [0, 51].

A restrição (12) garante que o parâmetro de quantização de *frames* tipo B (QB), varie dentro do intervalo de números inteiros positivos [0, 51].

Tabela 8 – máximo *frame rate* de acordo com o nível do H.264.

Nível (Profile Level)	QCIF 176x144	CIF 352x288	525SD 720x480	720pHD 1280x720	1080HD 1920x1088
1	15	-	-	-	-
1b	15	-	-	-	-
1.1	30,3	7,6	-	-	-
1.2	60,6	15,2	-	-	-
1.3	120	30	-	-	-
2	120	30	-	-	-
2.1	172	50	-	-	-
2.2	172	51,1	15	-	-
3	172	102,3	30	-	-
3.1	172	172	80	30	-
3.2	172	172	161	60	-
4	172	172	172	68,3	30
4.1	172	172	172	68,3	30
4.2	172	172	172	145,1	63,8
5	172	172	172	163,8	72,3
5.1	172	172	172	172	120,5

Fonte: ITU-T (2007).

A restrição (13) garante que a quantidade de frames tipo B dentro de um GOP (Group Of Picture) de tamanho igual a 10 frames, varie dentro de um intervalo de números inteiros positivos [0, 8].



A restrição (14) garante que o peso não tendencioso da variável *bit rate* (BR) seja maior que zero.

A restrição (15) garante que o peso não tendencioso da variável *frame rate* (FR) seja maior que zero.

A restrição (16) garante que o peso não tendencioso da variável referente ao parâmetro de quantização de *frames* tipo I (QI) seja maior que zero.

A restrição (17) garante que o peso não tendencioso da variável referente ao parâmetro de quantização de *frames* tipo P (QP) seja maior que zero.

A restrição (18) garante que o peso não tendencioso da variável referente ao parâmetro de quantização de *frames* tipo B (QB) seja maior que zero.

A restrição (19) garante que o peso não tendencioso da variável referente a quantidade de frames tipo B em um GOP (PF) seja maior que zero.

### 7.2.2 Solução Não Tendenciosa

A solução não tendenciosa (SNT) é aquela em que cada parcela da FO contribui da mesma forma que as demais, sem haver privilégio de alguma das variáveis de decisão. Para obter uma SNT, os pesos das variáveis de decisão da FO são calibrados de forma a normalizar os valores que cada parcela pode assumir.,

Para calcular os pesos da SNT, o aplicativo SMC foi executado 100 vezes onde foram atribuídos valores aleatórios inteiros de 0 a 100 aos pesos, seguindo uma distribuição normal. A BT usou a seguinte configuração: *nbmax* igual a 100 (critério de parada) e lista tabu de tamanho 7. O AG utilizou uma taxa de *crossover* igual a 0.8 e taxa de mutação igual a 0.2. A mesma solução inicial foi usada em todas as execuções.

Após 100 rodadas do SMC, tirou-se uma média dos valores assumidos pelas variáveis de decisão, onde a variável de maior média teve seu peso normalizado no valor 1 e o peso das demais variáveis foi calculado dividindo-se a maior média encontrada pelas médias de cada parâmetro, sendo o resultado desta divisão então arredondado e atribuído como peso não tendencioso a cada uma das variáveis de

decisão da FO. Os pesos não tendenciosos encontrados foram:  $\alpha_1 = 7$ ,  $\alpha_2 = 1$ ,  $\alpha_3 = 32$ ,  $\alpha_4 = 24$ ,  $\alpha_5 = 21$ ,  $\alpha_6 = 122$ , que correspondem respectivamente aos pesos do *bit rate*, *frame rate*, quantização de quadros I, quantização de quadros P, quantização de quadros B, quantidade de quadros B em um GOP.

### 7.3 Arquitetura Híbrida do Modelo SMC

Uma arquitetura híbrida é uma combinação de técnicas para se obter melhores resultados. Neste caso serão usadas as metaheurísticas Busca Tabu e Algoritmo Genético para compor a arquitetura do modelo proposto denominado pela sigla SMC a qual é mostrada na Figura 14.

As pesquisas estão se voltando ao uso de técnicas híbridas (Burk *et al*, 1998, 2001a; Gómez *et al*, 2005; Gonçalves *et al*, 2005), pois elas têm apresentado melhores resultados, visto que exploram as vantagens de cada método aplicado.

O modelo computacional proposto na Figura 14 tem como finalidade encontrar dentro de um espaço de busca, a melhor configuração de parâmetros para CODEC H.264 mostrado na Figura 12. A ideia principal é que as duas metaheurísticas, BT e AG trabalhem em conjunto para encontrar uma solução ótima para o problema de configuração de parâmetros do CODEC H.264/AVC do SBTVD.

A arquitetura híbrida do SMC é composta por quatro módulos: O primeiro módulo configura o SMC e recebe a solução inicial; o segundo módulo executa a Busca Tabu; o terceiro módulo executa o Algoritmo Genético e o quarto e último módulo é o CODEC H.264 propriamente dito que será configurado de acordo com a melhor solução encontrada pelo SMC.

O modelo SMC proposto neste trabalho é robusto no sentido de poder pesquisar a melhor configuração do CODEC H.264 para vários tipos de vídeo codificados nos diferentes perfis do padrão H.264 em todos os seus níveis.

Na literatura foram encontrados estudos (Yuelei *et al*, 2000), (Yasakethu *et al*, 2008) onde se aplica o Algoritmo Genético em módulos específicos de um CODEC H.264. Yuelei *et al* (2000) aplica um AG no módulo de Estimação de Movimento do

CODEC enquanto que em Yasakethu et al (2008) encontramos um AG aplicado ao módulo de Quantização. Em Koumaras et al (2005) foi estudado o comportamento *do bit rate* e *frame rate* para vídeos do tipo CIF e QCIF a fim de criar uma métrica objetiva para avaliar a qualidade de vídeo percebida.

Não foi encontrado na literatura a aplicação de Busca Tabu à módulos específicos do CODEC H.264.

O modelo proposto tem como objetivo fazer uma simulação do comportamento de seis dos principais parâmetros de configuração do H.264 e a partir disso gerar uma métrica objetiva que avalie o vídeo codificado em função de seu ruído (degradação da imagem), medido pelo PSNR, e da sua compressão, medida pelo tamanho final do vídeo codificado. O principal objetivo da métrica proposta, representada pela FO do modelo, é buscar o equilíbrio entre o ruído ou grau de degradação do vídeo codificado e seu grau de compressão e com isso, garantir um bom desempenho para o CODEC H.264.

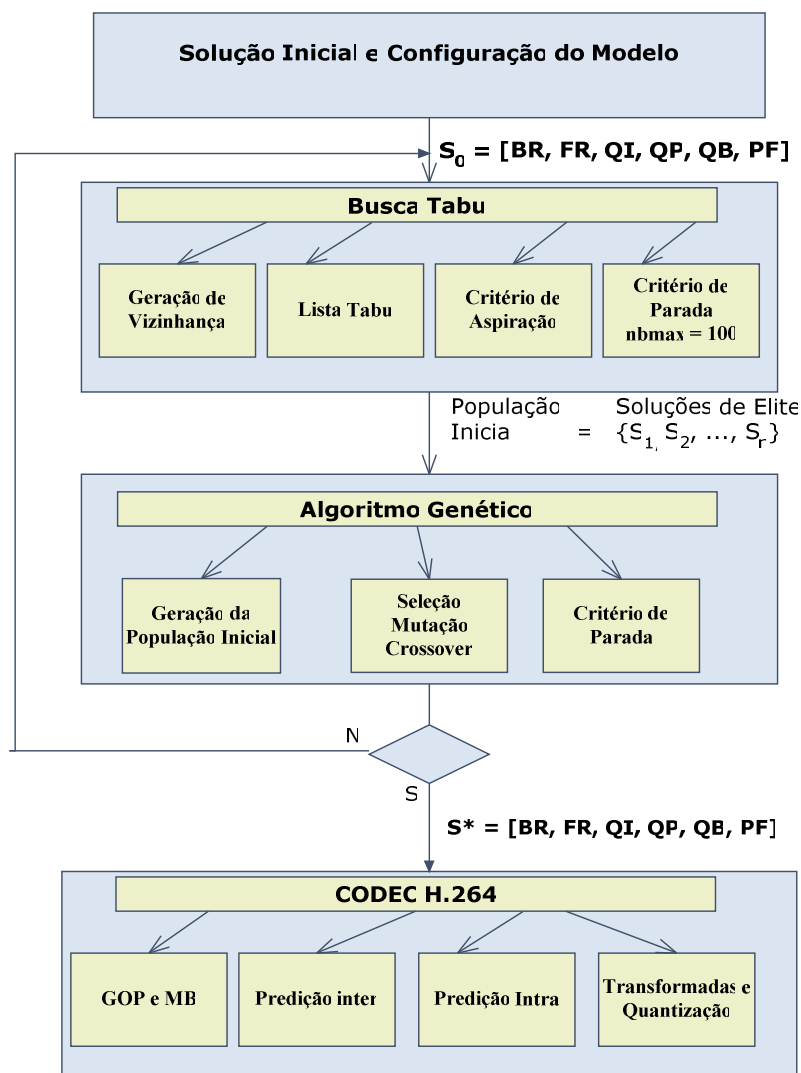


Figura 14 – Modelo Computacional do SMC.

Primeiramente é fornecido para o modelo uma solução inicial viável  $S_0$  e o tipo de vídeo que será simulado. Este código é o mesmo apresentado na Tabela 16 e que representa o formato do vídeo, o perfil e o nível do H.264 que devem ser simulados pelo algoritmo. A solução inicial também é identificada por esse código de vídeo.

As Soluções Iniciais de cada tipo de vídeo apresentadas na Tabela 16 são mostradas na Tabela 17. Cada Solução Inicial é obtida a partir de testes de codificação de um vídeo feitos pela equipe do projeto DigConv (UNISINOS, 2008a) utilizando para isso o CODEC H.264 desenvolvido por eles. A equipe configura os parâmetros do CODEC usando valores padrões de configuração obtidos na norma do padrão H.264 (ITU-T, 2007). Dentre os diversos parâmetros configurados estão

os seis parâmetros que fazem parte da Solução Inicial a ser fornecida ao SMC, que são: BR, FR, QI, QP, QB e PF.

A equipe utiliza a JIGA de testes do projeto para efetuar os testes completos de codificação de um vídeo e armazena os resultados obtidos, dentre eles, o valor do PSNR alcançado. Em resumo, esses testes consistem em configurar o CODEC H.264 da melhor forma possível com o objetivo de se alcançar um bom PSNR e uma boa compressão de vídeo. O vídeo que alcançou o melhor PSNR durante os testes foi aquele cujo CODEC foi melhor configurado. Esta configuração então é a solução inicial fornecida ao SMC.

### 7.3.1 Módulo da Busca Tabu

A Busca Tabu vai operar sobre o conjunto de parâmetros do CODEC H.264 da solução inicial  $S_0$ , a fim de gerar um conjunto das melhores soluções viáveis, chamadas soluções de elite. As soluções de elite são aquelas que apresentaram os valores mais altos na Função Objetivo. A BT utiliza como Função Objetivo a equação (7) já apresentada no item 7.2.1.

A quantidade de soluções de elite geradas pela a BT é de no máximo 20 soluções. Estas soluções são enviadas ao AG para formarem sua população inicial. A BT pode atingir seu critério de parada ( $nbmax=100$ ) sem contudo ter conseguido gerar 20 soluções de elite. Caso isso ocorra ela vai enviar ao AG as soluções de elite geradas até o momento e este vai complementar sua população inicial de forma aleatória. O módulo de geração de vizinhança da BT vai gerar inicialmente 50 vizinhos. Todos os vizinhos serão gerados a partir da perturbação de todas as variáveis de decisão. Esse tipo de geração de vizinhança vai dar a BT uma característica de diversificação, conforme será visto mais adiante na análise dos experimentos.

A BT utiliza uma Lista Tabu composta inicialmente de no máximo 7 movimentos Tabu. Essa lista guarda uma solução completa, ou seja, os valores de todas as variáveis de decisão que fazem parte de uma mesma solução, fazendo com que essa solução seja considerada um movimento tabu.

O módulo de critério de aspiração da BT prevê que será aceito um movimento, mesmo que tabu, se ele melhorar o valor da Função Objetivo global. O critério de parada utilizado pela BT é o *nbmax* de tamanho igual a 100. Quando a BT atinge seu critério de parada então ela passa o controle para o Algoritmo Genético e envia a ele o conjunto de soluções de elite.

### 7.3.2 Módulo do Algoritmo Genético

O módulo Algoritmo Genético recebe como entrada o conjunto de soluções de elite gerados pela BT para ser sua população inicial. Caso a BT tenha enviado ao AG uma quantidade de soluções de elite menor que 20, então o módulo de geração de vizinhança do AG complementa sua população inicial gerando aleatoriamente quantos indivíduos forem necessários para se chegar a uma população inicial de 20 indivíduos. O AG utiliza como função de *fitness* a equação (7) já apresentada no item 7.2.1. Esta função avalia o grau de aptidão de um indivíduo da população, neste caso, quanto maior o *fitness*, melhor a qualidade da solução encontrada. O cromossomo é representado pelo conjunto de seis parâmetros que são os mesmos parâmetros de configuração do CODEC H.264 utilizados no modelo SMC, que são: BR, FR, QI, QP, QB e PF.

O módulo de seleção do AG trabalha com a seleção por torneio conforme descrita na seção 6.4.2 e desta seleção sempre serão escolhidos dois pais.

O módulo de cruzamento do AG utiliza uma taxa 0,8. É usado o cruzamento aritmético para as variáveis pertencente ao conjunto dos números reais, que são o *bit rate* e o *frame rate*, conforme descrito na seção 6.5.2. Para as variáveis pertencentes ao conjunto dos números inteiros positivos, que são, QI, QP, QB e PF, foi usado o cruzamento de um ponto de corte, conforme descrito na seção 6.5.1. Em resumo, teve-se que usar dois tipos de *crossover*, o aritmético e o de um ponto de corte em função das variáveis de decisão pertencerem a diferentes conjuntos numéricos, umas pertencem ao conjunto dos números inteiros positivos e outras ao conjunto dos números reais positivos.

A Tabela 9 mostra o modelo do cromossomo, onde as variáveis que pertencem ao mesmo tipo de conjunto numérico ficaram próximas na hora de montar o

cromossomo. Isso foi feito a fim de facilitar o *crossover* de um ponto de corte usado entre as variáveis de decisão pertencentes ao conjunto dos números inteiros:

Tabela 9 – *Crossover* aritmético e *crossover* de um ponto

Cromossomo 1	BR	FR		QI		QP		QB		PF	
	64,53		33,22		20		28		30		2
Cromossomo 2	85,34		86,17		21		20		30		1
Filho 1	80,13		72,93		20		28		30		1
Filho 2	69,73		46,45		21		20		30		2

Para o *crossover* aritmético usado entre as variáveis BR e FR foram utilizadas as equações (7) e (8) sendo  $\alpha = 0,25$ . Para o *crossover* de um ponto de corte utilizado

entre as variáveis QI, QP, QB e PF, foi usado como ponto de corte o valor de variável aleatória inteira dentro do intervalo [1,3].

No exemplo da Tabela 9 o ponto de corte sorteado foi igual a 2, o que significa que este ponto ficou na posição entre as variáveis QP e QB. Logo, o algoritmo executou o seguinte cruzamento: o primeiro filho herdou do cromossomo 1 os parâmetros QI e QP (valores 20 e 28 respectivamente) e herdou do cromossomo 2 os parâmetros QB e PF (valores 30 e 1, respectivamente). O segundo filho herdou do cromossomo 1 os parâmetros QB e PF (valores 30 e 2 respectivamente) e herdou do cromossomo 2 os parâmetros QI e QP (valores 21 e 20 respectivamente).

O módulo de mutação do AG utiliza a mutação uniforme conforme descrito na seção 6.6.1 com uma taxa de mutação de 0,2. É sorteado aleatoriamente um dos pais para sofrer o processo de mutação.

O critério de parada do AG é o número de gerações igual a 100. Quando o AG atinge esse número de gerações o modelo SMC compara se houve uma melhora na função *fitness* em relação a melhor função objetivo da BT, caso tenha havido essa melhora então o SMC aciona novamente a BT passando para ela como solução inicial a função *fitness* do AG. Caso não tenha ocorrido melhora na FO então o SMC

termina a execução do algoritmo e adota a função objetivo da BT como sendo a melhor solução encontrada pelo modelo que é chamada de  $S^*$ . Essa solução  $S^*$  será então usada nas configurações do CODEC H.264/AVC.

## **8 VALIDAÇÃO DO MODELO**

A validação do modelo foi realizada com base em experimentos que analisaram o comportamento das variáveis de decisão e qual a influência que uma variável exerce sobre outra e para isso observou-se o comportamento da Função Objetivo. Tanto os comportamentos quanto o tipo de influência das variáveis de decisão foram comparados com a literatura. Ao final dos experimentos o modelo foi validado e comprovou-se que seu comportamento está de acordo com a literatura.

Foram feitos experimentos com as variáveis de decisões para validar o modelo, onde procurou-se analisar o comportamento das variáveis de decisão na FO e a influencia que uma variável de decisão exerce sobre as demais na FO.

### **8.1 Primeira Etapa dos Experimentos de Validação do Modelo**

Neste experimento foram feitas 300 execuções completas do SMC para cada uma das variáveis de decisão, sendo que a cada trinta execuções foram calculados os valores médios atingidos pelas variáveis e uma delas teve seu peso incrementado em 20 unidades, para que seu grau de contribuição na FO aumentasse a cada 30 execuções. Como o experimento foi feito para cada uma das seis variáveis de decisão que fazem parte da FO, então ao todo, o algoritmo foi executado 1800 vezes para este experimento.

O objetivo deste experimento é comprovar a influência que uma variável de decisão exerce sobre a FO e as demais variáveis quando a mesma tem um maior grau de contribuição na FO em função do aumento de seu peso.



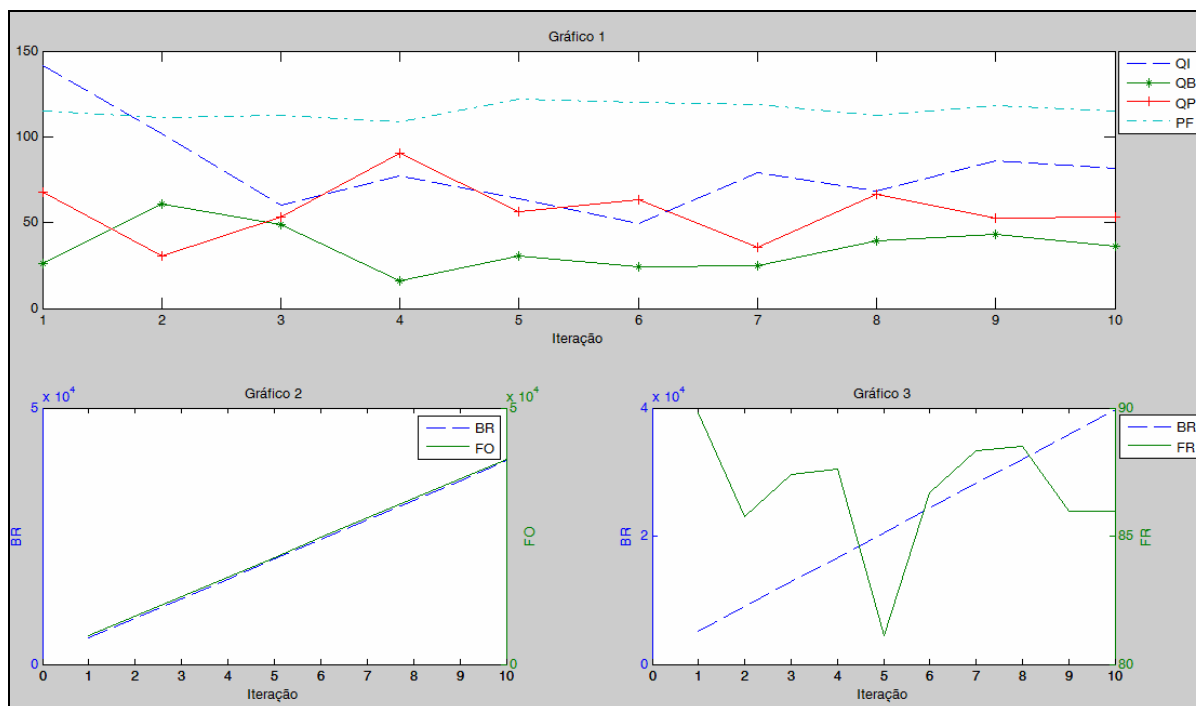


Figura 15 – Comportamento das variáveis de decisão à medida que o peso de BR aumenta na FO.

A Figura 15 mostra os gráficos referentes às curvas de comportamentos da FO e das variáveis de decisão estudadas em função do aumento da contribuição de BR na FO.

O Gráfico 1 (Figura 15) mostra o comportamento das variáveis de decisão QI, QB, QP e PF em relação a BR. O Gráfico 2 (Figura 15) mostra o comportamento da FO em relação a BR. O Gráfico 3 (Figura 15) mostra o comportamento do FR em relação a BR.

O Gráfico 1 (Figura 15) mostra que as variáveis QP e QB tendem a ter um comportamento inversamente proporcional e a variável PF tem um comportamento diretamente proporcional a QB. Segundo as normas do H.264/AVC (ITU-T, 2007), quadros tipo B são os que sofrem as maiores compressões, quanto mais quadros tipo B (variável PF) estiverem presentes em um GOP, mais compressão um vídeo sofrerá e isto se reflete no aumento de QB, que é o fator de compressão de quadros B. Nota-se que para compensar essa perda de qualidade de imagem, devido à alta compressão de quadros B, o SMC tende a diminuir a compressão de quadros tipo P (QP), pois são quadros que servem de referência para quadros tipo B no momento da reconstituição do vídeo (ITU-T, 2007). Com este procedimento o SMC tenta compensar a perda da qualidade da imagem devido ao aumento de quadros tipo B.

Outro comportamento observado neste gráfico é que os valores de QP, QB e QP, que são variáveis diretamente relacionadas à compressão da imagem, não alcançam altos valores. Como nesse cenário temos um *bit rate* em constante crescimento, então ele tende a inibir o aumento exagerado desses parâmetros. Esse comportamento vai gerar um equilíbrio na codificação da imagem, pois o *bit rate* é inversamente proporcional às variáveis QI, QP e QB em se tratando de conseguir uma melhora na qualidade de imagem.

O Gráfico 2 (Figura 15) mostra que à medida que o *bit rate* aumenta, também aumenta a qualidade da imagem que se reflete no aumento da FO. Segundo (Yasakethu *et al*, 2008) quando se alcançam altas taxas de *bit rate* é mais fácil dividir um MB em partições menores o que conseqüentemente vai contribuir para uma melhor qualidade da imagem. Em outras palavras o aumento do *bit rate* vai gerar um aumento na FO, pois quanto maior a FO maior será a qualidade do vídeo codificado pelo H.264. Entretanto, de acordo com (Koumaras *et al*, 2005) a melhora da qualidade de vídeo não é significativa para *bit rates* mais altos que um limite específico, pois a variação do nível de qualidade em relação ao *bit rate* é uma função não linear crescente com inclinação de acordo com a dinâmica do vídeo.

O Gráfico 3 (Figura 15) mostra que à medida que BR aumenta, FR tende a sofrer uma oscilação em seus valores. Esse comportamento pode ser comprovado em (Ries *et al*, 2007) onde foi observado que dependo do conteúdo do vídeo, as taxas de *frame rate* podem aumentar ou diminuir em relação ao aumento da taxa de *bit rate*. Nos experimentos feitos por (Ries *et al*, 2007), vídeos panorâmicos receberam uma melhor avaliação na qualidade de vídeo quando o *frame rate* caiu e vídeos com sequencias dinâmicas foram avaliados melhor à medida em que subiam as taxas de *frame rate* e *bit rate*. Neste gráfico também podemos observar que na iteração 5 o *frame rate* sofre uma queda brusca, isso acontece em função de PF atingir seu valor médio máximo, o que significa dizer que quando temos uma maior quantidade de quadros tipo B (PF), que são os quadros que sofrem as maiores compressões (ITU-T, 2007), o *bit rate* tende a cair devido a esses tipos de quadros levarem um maior tempo em sua codificação.

A Tabela 10 mostra a média e o desvio-padrão da FO e das variáveis de decisão BR, FR, QI, QP, QB e PF. Cada linha da tabela corresponde à média de 30

execuções do SMC para cada peso associado à variável BR. Por exemplo, na iteração 1 da tabela, podemos ver que BR possui peso igual a 17, para este caso, foi executado 30 vezes o algoritmo a fim de obter as 30 melhores FOs e a partir disso calcular uma média da FO e de suas variáveis de decisão. Esta tabela deu origem aos gráficos da Figura 15.

Tabela 10 – Médias obtidas à medida que o peso de BR aumenta na FO.

Iter.	FO		BR			FR		QI		QP		QB		PF	
	Média	$\sigma$	Média	$\sigma$	Peso	Média	$\sigma$	Média	$\sigma$	Média	$\sigma$	Média	$\sigma$	Média	$\sigma$
1	5580,45	85,21	5139,68	39,10	17	89,81	13,25	141,51	125,13	67,86	89,13	26,36	50,33	115,22	20,91
2	9378,80	69,28	8988,41	41,22	37	85,76	14,34	101,93	111,98	30,41	47,29	60,87	80,10	111,43	28,25
3	13195,66	63,96	12833,40	30,60	57	87,41	12,27	60,09	83,08	53,45	68,59	48,8	67,28	112,51	24,89
4	17031,71	45,81	16651,97	51,35	77	87,61	11,38	76,91	96,17	90,46	89,88	16,32	35,79	108,44	27,92
5	20866,86	63,91	20512,54	24,77	97	81,104	19,01	63,86	78,61	56,50	84,27	30,82	55,13	122,00	0,00
6	24695,78	54,92	24351,64	28,53	117	86,70	11,11	49,59	82,19	63,49	68,21	24,39	44,17	119,97	11,14
7	28539,86	60,42	28193,17	26,64	137	88,32	11,68	79,10	92,81	35,43	56,42	24,89	41,74	118,95	16,71
8	32411,56	77,93	32036,43	17,79	157	88,50	7,68	68,14	88,33	66,48	86,42	39,50	55,55	112,51	24,89
9	36240,54	82,80	35854,89	40,93	177	85,97	11,92	86,01	95,15	52,71	63,11	43,02	69,37	117,93	15,48
10	40081,34	83,85	39709,56	34,2	197	85,96	9,79	81,31	99,23	53,10	71,43	36,53	61,13	114,88	22,20

Os valores em destaque na Tabela 10 correspondem aos valores médios máximos e mínimos alcançados pela FO e suas variáveis de decisão quando BR tem um maior grau de contribuição na FO. Se compararmos as colunas de QP e QB da Tabela 10, podemos notar que os valores mínimos e máximos dessas duas variáveis coincidem de forma inversa, ou seja, quando QP atinge seu maior valor médio, QB atinge seu menor valor médio e vice-versa. Isso acontece devido ao SMC tentar encontrar um equilíbrio entre a compressão dos quadros tipo P (QP) e tipo B (QB). De acordo com (ITU-T, 2007), quadros B buscam como referência quadros P, ou seja, quanto menos se comprime um quadro P mais comprimido pode ser um quadro B, pois isso vai contribuir para uma melhor qualidade da imagem.

A Figura 16 mostra os gráficos correspondentes aos comportamentos da FO e de suas variáveis de decisão quando FR tem um maior grau de contribuição na FO. A Tabela 11 mostra a média e o desvio-padrão da FO e das variáveis de decisão, BR, FR, QI, QP, QB e PF, que deram origem a esses gráficos. Essa tabela segue os mesmos moldes da Tabela 10, com a diferença de que neste caso o peso de FR está sendo incrementado.

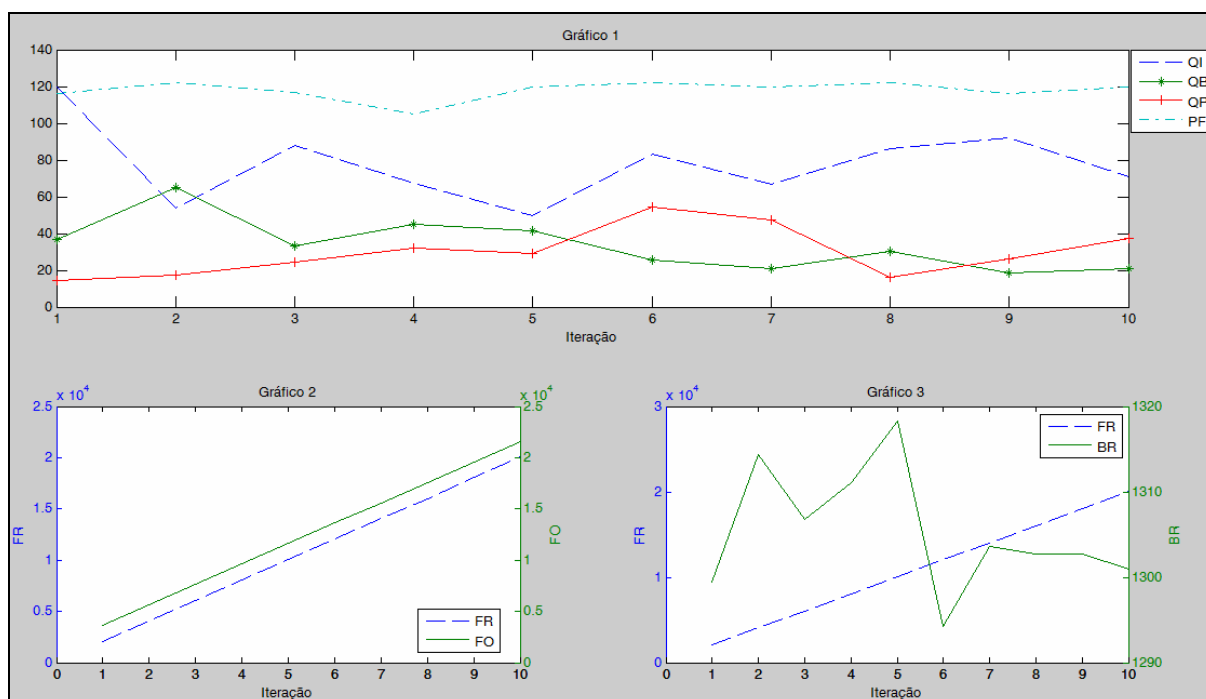


Figura 16 – Comportamento das variáveis de decisão à medida que o peso de FR aumenta na FO.

Pode-se notar no Gráfico 1 (Figura 16) que o comportamento das variáveis QI, QP, QB e PF permaneceram os mesmos observados no Gráfico 1 (Figura 15).

Tabela 11 – Médias obtidas à medida que o peso de FR aumenta na FO.

Iter.	FO		BR		FR		Peso	QI		QP		QB		PF	
	Média	$\sigma$	Média	$\sigma$	Média	$\sigma$		Média	$\sigma$	Média	$\sigma$	Média	$\sigma$	Média	$\sigma$
1	3643,37	59,21	1299,43	33,80	2056,80	48,42	20	119,84	112,55	14,36	30,94	37,04	66,15	115,90	18,61
2	5640,62	75,01	1314,31	15,72	4068,09	15,42	40	53,60	88,67	17,55	43,46	65,07	61,61	122,00	0,00
3	7621,59	65,87	1306,84	26,99	6052,41	28,62	60	88,09	96,29	24,12	43,72	33,22	47,53	116,92	19,76
4	9606,92	92,46	1311,06	19,36	8046,38	52,29	80	67,15	92,57	32,22	61,93	44,99	69,44	105,12	35,16
5	11611,65	87,21	1318,33	15,73	10053,19	39,98	100	49,70	65,87	28,99	50,47	41,47	63,94	119,97	11,14
6	13633,68	99,16	1294,27	32,7	12053,40	33,24	120	83,47	99,79	54,68	78,20	25,87	47,58	122,00	0,00
7	15605,19	62,40	1303,59	28,12	14046,35	42,10	140	66,98	98,62	47,30	69,94	21,01	43,04	119,97	11,14
8	17588,78	96,98	1302,74	31,48	16031,25	44,68	160	86,17	99,59	16,37	27,74	30,25	52,03	122,00	0,00
9	19609,70	77,14	1302,74	29,94	18054,25	31,75	180	92,32	89,51	25,98	51,34	18,49	32,60	115,90	23,21
10	21596,94	98,95	1300,94	30,20	20047,17	36,68	200	70,70	93,72	37,12	58,78	21,04	46,52	119,97	11,14

Pode-se observar na Tabela 11 que os valores médios mínimos e máximos que a FO alcançou ficaram abaixo dos valores alcançados quando BR tinha um maior grau de contribuição na FO, conforme mostrado na Tabela 10. Isso nos leva a concluir que BR contribui de maneira mais efetiva para o aumento da FO e consequentemente para uma melhor qualidade de imagem.

A Figura 17 mostra os gráficos correspondentes aos comportamentos da FO e de suas variáveis de decisão quando QI tem um maior grau de contribuição na FO.

A Tabela 12 mostra a média e o desvio-padrão da FO e das variáveis de decisão, BR, FR, QI, QP, QB e PF, que deram origem a esses gráficos. Essa tabela segue os mesmos moldes da Tabela 10, com a diferença de que neste caso o peso de QI está sendo incrementado.

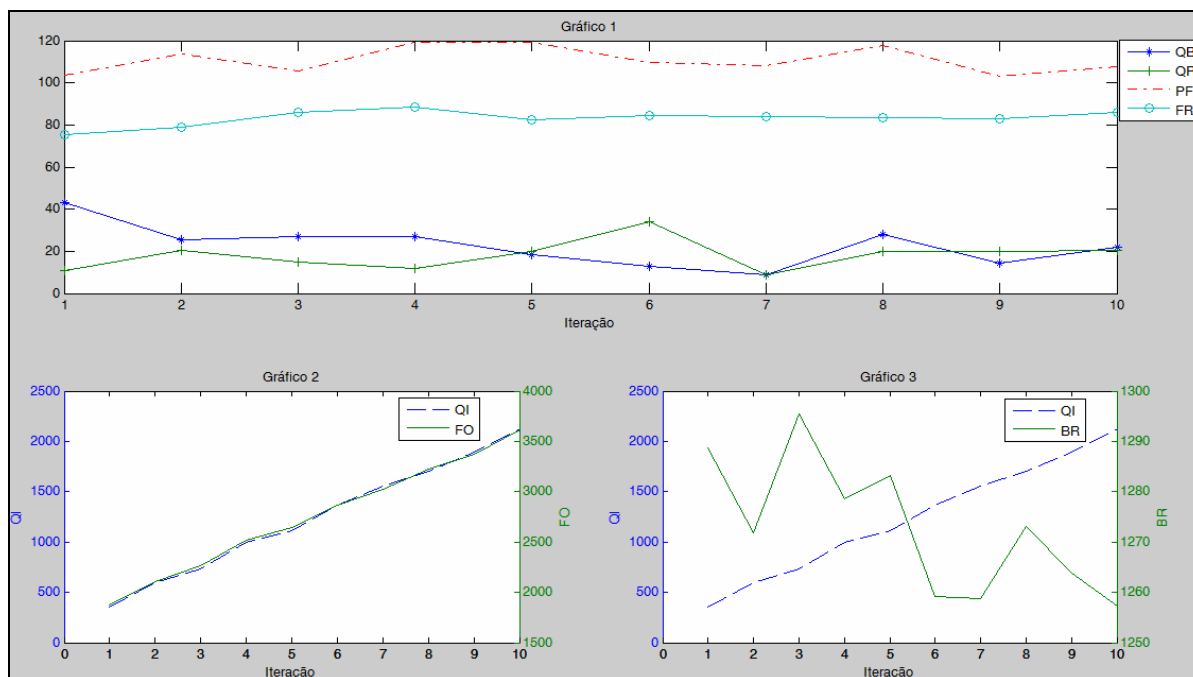


Figura 17 – Comportamento das variáveis de decisão à medida que o peso de QI aumenta na FO.

O Gráfico 1 (Figura 17) mostra o comportamento das variáveis de decisão QB, QP, PF e FR em relação a QI. O Gráfico 2 (Figura 17) mostra o comportamento da FO em relação a QI. O Gráfico 3 (Figura 17) mostra o comportamento do BR em relação a QI.

O Gráfico 1 (Figura 17) mostra que quando QI contribui em maior grau na FO faz com que QP e QB se comportem de maneira inversamente proporcional, o que vai contribuir para um melhor qualidade da imagem.

Tabela 12 – Médias obtidas à medida que o peso de QI aumenta na FO.

Iter	FO		BR		FR		QI			QP		QB		PF	
	Média	$\sigma$	Média	$\sigma$	Média	$\sigma$	Média	$\sigma$	Peso	Média	$\sigma$	Média	$\sigma$	Média	$\sigma$
1	1870,69	104,37	1288,61	50,01	75,21	21,76	349,51	170,19	52	10,66	14,09	43,00	74,66	103,70	33,92
2	2110,84	112,06	1271,90	48,69	78,91	17,52	600,32	84,54	72	20,43	41,92	25,42	51,28	113,87	25,24
3	2263,23	122,29	1295,40	37,58	85,79	11,69	734,69	112,23	92	14,87	30,50	26,75	47,89	105,73	30,39
4	2521,15	101,88	1278,61	60,08	88,39	10,45	995,86	74,65	112	11,91	17,72	27,09	48,95	119,29	14,85
5	2641,06	146,55	1283,29	58,73	82,20	12,89	1117,85	115,22	132	19,79	40,28	18,65	30,73	119,29	14,85
6	2865,54	118,97	1259,19	84,56	84,34	15,13	1365,77	100,39	152	34,08	57,59	12,69	31,30	109,46	28,96
7	3023,17	156,44	1258,66	74,36	83,96	9,97	1554,73	109,58	172	8,76	17,70	8,96	16,17	108,11	28,82
8	3227,92	132,87	1273,08	51,34	83,49	14,83	1705,31	123,21	192	20,00	41,45	28,12	53,38	117,93	22,27
9	3373,59	188,28	1263,71	74,27	82,93	10,92	1889,66	160,73	212	19,73	32,50	14,53	36,80	103,02	29,70
10	3620,18	152,29	1257,25	74,84	85,85	13,87	2127,10	131,35	232	20,51	34,50	21,70	38,83	107,77	33,41

O Gráfico 2 (Figura 17) mostra que QI está contribuindo para o aumento da FO, porém essa contribuição fica abaixo das contribuições de BR e FR vistas anteriormente. Isso ocorre devido ao fato de que quanto maior a compressão de um quadro, menor será a qualidade da imagem, por esse motivo o SMC não encontra valores altos para a FO.

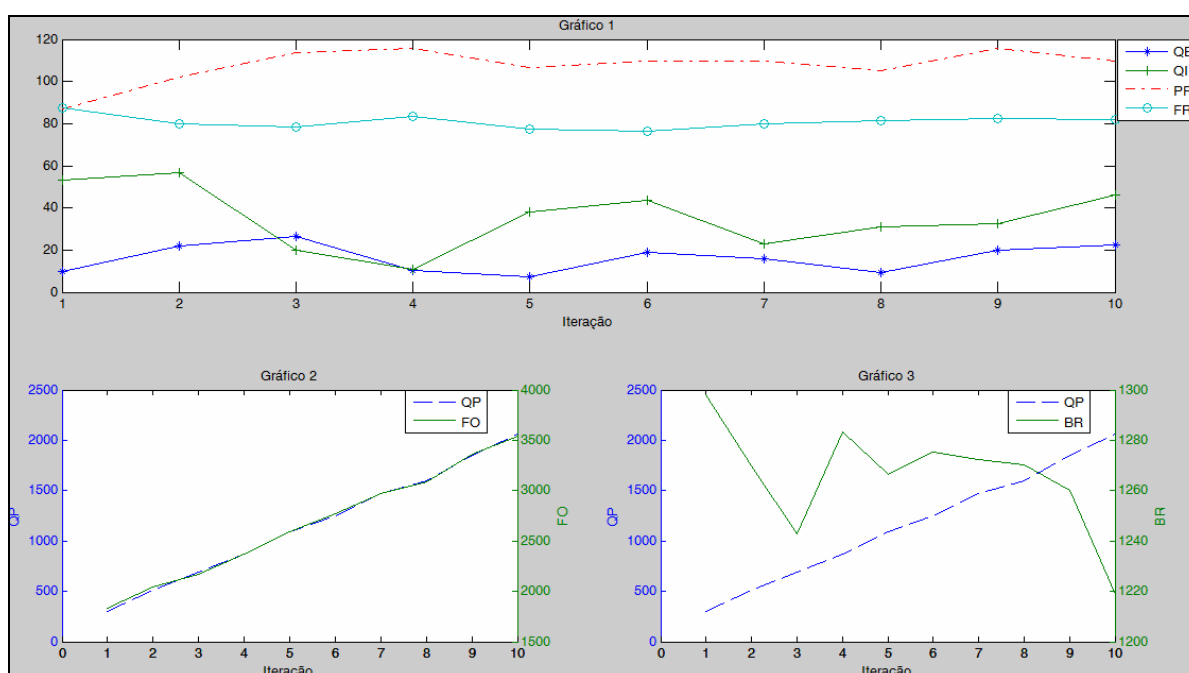


Figura 18 – Comportamento das variáveis de decisão à medida que o peso de QP aumenta na FO.

O Gráfico 3 (Figura 17) mostra que BR permanece em seu intervalo médio de valores máximos enquanto QI está em seu intervalo médio de valores mínimos e vice-versa. Isso ocorre porque o SMC tenta compensar o aumento do *bit rate* por

uma baixa compressão e vice-versa e com isso chegar a uma melhor qualidade da imagem.

A Figura 18 mostra os gráficos correspondentes aos comportamentos da FO e de suas variáveis de decisão quando QP tem um maior grau de contribuição na FO. A Tabela 13 mostra a média e o desvio-padrão da FO e das variáveis de decisão, BR, FR, QI, QP, QB e PF, que deram origem a esses gráficos. Essa tabela segue os mesmos moldes da Tabela 10, com a diferença de que neste caso o peso de QP está sendo incrementado.

O Gráfico 1 (Figura 18) mostra o comportamento das variáveis de decisão QB, QI, PF e FR em relação a QP. Nele pode-se observar que quando QP aumenta seu grau de contribuição na FO, isso faz com que o intervalo de médias de QI aumente e o intervalo de médias de QB diminua (ver Tabela 13). Esse comportamento nos diz que o SMC permite aumentar a compressão de quadros I à medida que QP aumenta, porém não deixa aumentar a compressão de quadros B. Como já foi dito anteriormente, quadros B usam quadros P e I como quadros de referência e se a compressão desses dois quadros aumenta então é aceitável que a compressão dos quadros B seja menor, pois isso vai contribuir para uma melhor qualidade da imagem.

Tabela 13 – Médias obtidas à medida que o peso de QP aumenta na FO.

Iter	FO		BR		FR		QI		QP			QB		PF	
	Média	$\sigma$	Média	$\sigma$	Média	$\sigma$	Média	$\sigma$	Média	$\sigma$	Peso	Média	$\sigma$	Média	$\sigma$
1	1833,35	68,81	1298,24	33,10	87,56	13,07	53,14	94,14	297,53	126,64	54	9,85	22,47	87,03	45,67
2	2043,01	122,32	1269,68	78,28	79,80	22,42	56,47	88,44	513,14	78,53	74	21,92	50,72	102,01	34,55
3	2172,05	101,09	1242,86	68,97	78,53	18,35	19,72	47,16	690,83	103,85	94	26,24	45,00	113,87	21,09
4	2371,46	151,14	1283,35	51,30	83,47	13,42	10,83	20,93	867,68	140,37	114	10,23	17,41	115,90	18,61
5	2592,81	123,41	1266,48	83,20	77,34	17,57	38,15	75,26	1096,88	92,57	134	7,55	20,53	106,41	29,09
6	2769,78	121,87	1275,21	63,99	76,32	22,82	43,41	74,14	1246,21	94,67	154	18,84	44,68	109,80	28,43
7	2973,90	133,33	1272,53	48,54	79,88	17,61	23,18	34,08	1472,69	124,96	174	15,81	31,22	109,80	28,43
8	3091,65	224,56	1270,24	55,65	81,51	15,25	31,23	55,36	1594,52	166,30	194	9,11	28,71	105,06	31,63
9	3361,77	116,90	1259,94	56,42	82,36	14,46	32,50	69,27	1851,12	117,31	214	19,96	44,42	115,90	18,61
10	3537,63	148,12	1218,83	87,97	81,77	13,86	46,36	65,79	2058,69	105,38	234	22,19	46,87	109,80	24,82

O Gráfico 2 (Figura 18) mostra o comportamento da FO em relação a QP, neste caso comprova-se mais uma vez que quando uma variável relacionada à compressão de quadros aumenta sua contribuição na FO, esta tende a não aumentar muito de valor devido o aumento da compressão de um quadro não favorecer a uma melhor qualidade de imagem.

O Gráfico 3 (Figura 18) mostra o comportamento do BR em relação a QP. Nele verifica-se mais uma vez que BR tende a cair quando se tem altas taxas de compressão.

A Figura 19 mostra os gráficos correspondentes às curvas de comportamentos da FO e de suas variáveis de decisão quando QB tem um maior grau de contribuição na FO. A Tabela 14 mostra a média e o desvio-padrão da FO e das variáveis de decisão, BR, FR, QI, QP, QB e PF, que deram origem a esses gráficos. Essa tabela segue os mesmos moldes da Tabela 10, com a diferença de que neste caso o peso de QB está sendo incrementado.

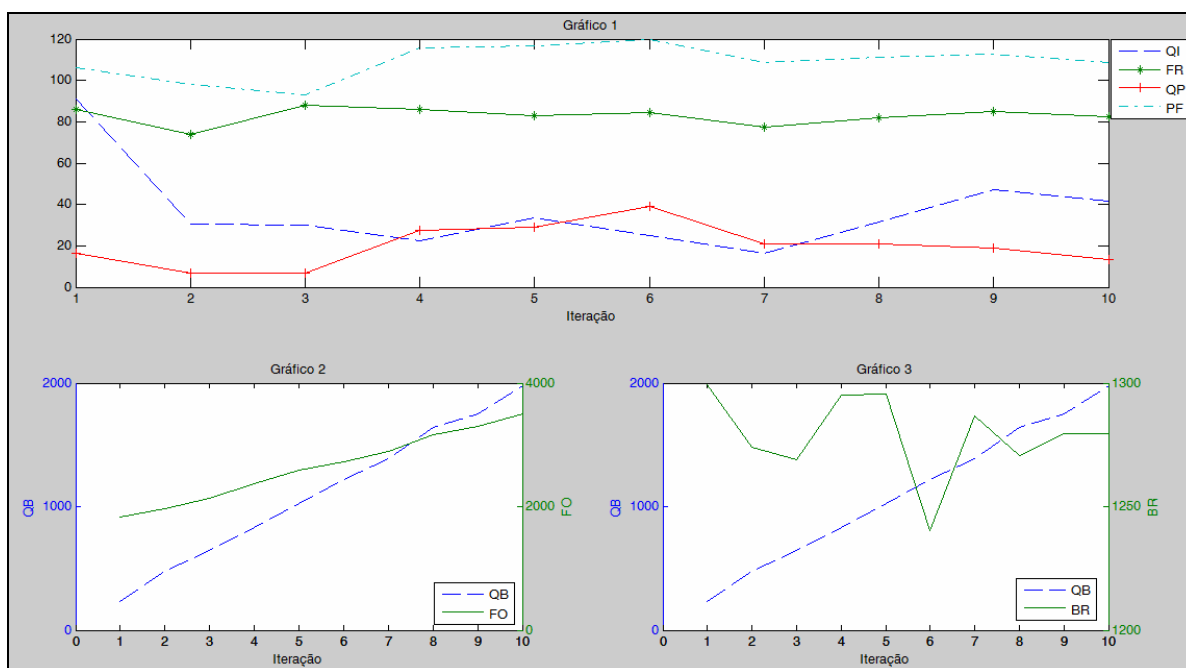


Figura 19 – Comportamento das variáveis de decisão à medida que o peso de QB aumenta na FO.

O Gráfico 1 (Figura 19) mostra que a faixa de valores médios de QP diminui devido ao aumento da contribuição de QB na FO. Isso comprova que o SMC tenta ganhar qualidade de imagem nos quadros P para compensar a grande perda de qualidade dos quadros B e assim tentar encontrar um equilíbrio na compressão desses quadros a fim de encontrar uma boa solução.



Tabela 14 – Médias obtidas à medida que o peso de QB aumenta na FO.

Iter	FO		BR		FR		QI		QP		QB			PF	
	Média	$\sigma$	Média	$\sigma$	Média	$\sigma$	Média	$\sigma$	Média	$\sigma$	Média	$\sigma$	Peso	Média	$\sigma$
1	1833,83	63,04	1299,37	55,05	85,82	11,20	91,11	121,23	16,24	39,53	235,22	141,27	41	106,07	29,93
2	1963,37	157,91	1273,95	63,32	73,67	25,45	30,39	69,61	6,91	12,55	480,18	113,03	61	98,28	37,60
3	2130,20	153,02	1268,91	69,29	87,92	15,28	30,08	72,38	6,64	21,41	643,80	144,87	81	92,86	42,78
4	2377,51	142,31	1295,00	49,51	85,84	10,02	22,26	60,63	27,71	59,20	830,80	143,84	101	115,90	18,61
5	2583,11	124,29	1295,36	50,13	83,06	12,56	33,33	50,97	29,17	55,52	1025,61	121,11	121	116,58	20,63
6	2724,89	115,36	1240,36	55,96	84,67	16,05	25,16	58,91	38,98	60,21	1215,76	107,21	141	119,97	11,14
7	2900,80	190,23	1286,92	55,66	77,46	16,45	16,34	28,69	20,75	41,95	1390,55	159,69	161	108,78	30,69
8	3160,65	102,35	1270,57	45,89	81,95	15,44	31,72	63,36	20,83	56,03	1644,41	68,52	181	111,16	28,50
9	3294,22	180,73	1279,69	45,06	84,82	14,74	47,01	102,07	19,16	40,13	1751,03	165,27	201	112,51	24,89
10	3502,70	122,60	1279,48	67,09	82,33	13,60	41,76	66,32	13,29	42,39	1977,39	140,70	221	108,44	27,92

Um ponto importante a ser notado se refere aos gráficos 1 e 3, ambos da Figura 19, onde na iteração 6 desses dois gráficos ocorre o ponto máximo atingido por PF no Gráfico 1 e o ponto mínimo alcançado por BR no Gráfico 3. Isto significa que quando o número de quadros B (PF) atinge seu maior valor médio e a compressão desse tipo de quadro (QB) está em crescimento, o *bit rate* atinge o seu valor médio mínimo, pois com uma grande quantidade de quadros B a altas taxas de compressão fazem com que o codificador demore mais para codificar um quadro, o que faz com que a taxa de *bit rate* tenha uma queda. Isso demonstra que o SMC está plenamente em conformidade com o comportamento dessas variáveis e para este caso ele vai encontrar um baixo valor para a FO.

O Gráfico 2 (Figura 19) comprova mais uma vez que o aumento da contribuição de uma variável de compressão na FO não favorece que a mesma alcance intervalos médios mais altos e assim consiga melhorar a qualidade da imagem.

A Figura 20 mostra os gráficos correspondentes às curvas de comportamentos da FO e de suas variáveis de decisão quando PF tem um maior grau de contribuição na FO. A Tabela 15 mostra a média e o desvio-padrão da FO e das variáveis de decisão, BR, FR, QI, QP, QB e PF, que deram origem a esses gráficos. Essa tabela segue os mesmos moldes da Tabela 10, com a diferença de que neste caso o peso de PF está sendo incrementado.

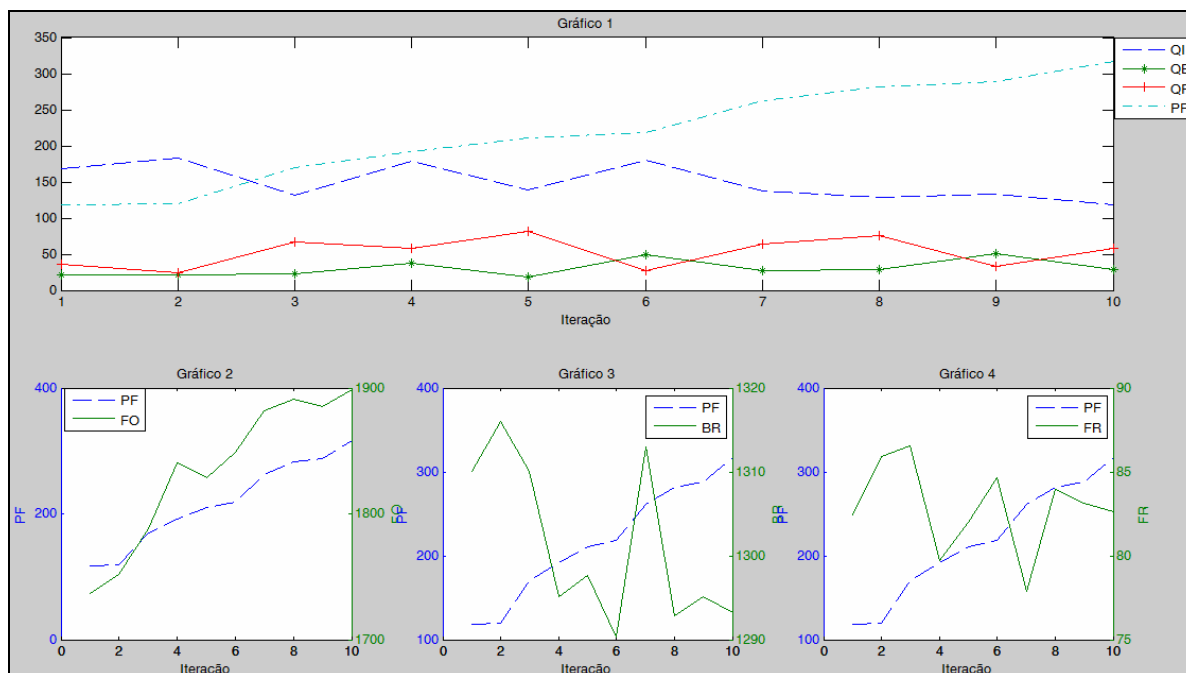


Figura 20 – Comportamento das variáveis de decisão à medida que o peso de PF aumenta na FO.

O Gráfico 1 (Figura 20) mostra os padrões de comportamento das variáveis QI, QB e QP a medida que PF aumenta seu grau de contribuição na FO. Podemos notar que QP e QB se comportam de maneira inversa conforme visto nos gráficos anteriores. QP e QI por sua vez também se comportam de maneira inversa seguindo a mesma lógica dos quadros P e B, ou seja, se quadros P usam como quadros de referencia os quadros I então quanto mais se comprimir um quadro P menos se deve comprimir um quadro I a fim de contribuir para uma melhor qualidade de imagem.

Tabela 15 – Médias obtidas à medida que o peso de PF aumenta na FO.

Iter	FO		BR		FR		QI		QP		QB		PF		Peso
	Média	$\sigma$	Média	$\sigma$	Média	$\sigma$	Média	$\sigma$	Média	$\sigma$	Média	$\sigma$	Média	$\sigma$	
1	1736,37	73,01	1310,00	33,34	82,43	14,25	168,72	108,57	36,02	65,51	21,25	49,52	117,94	41,40	142
2	1751,67	59,67	1315,96	19,47	85,93	9,49	183,81	96,11	24,65	46,99	21,61	49,60	119,70	43,53	162
3	1788,16	73,50	1310,08	41,21	86,54	14,17	131,62	113,79	66,55	81,20	23,48	49,54	169,87	31,46	182
4	1840,41	69,51	1295,18	31,71	79,74	18,03	177,99	120,65	58,07	79,73	37,53	57,17	191,90	30,82	202
5	1828,85	63,20	1297,61	41,39	81,98	15,78	138,29	128,51	81,53	91,59	18,54	41,89	210,90	33,87	222
6	1848,84	72,84	1290,31	45,81	84,68	14,94	180,14	112,27	26,89	47,46	49,03	75,86	217,80	49,23	242
7	1881,45	80,95	1313,00	29,66	77,88	17,21	137,98	101,28	63,83	82,79	26,77	51,97	262,00	0,00	262
8	1891,37	86,15	1292,80	52,75	83,93	12,94	128,34	118,24	76,11	95,23	28,19	54,23	282,00	0,00	282
9	1884,91	65,48	1295,10	53,62	83,15	16,17	133,75	126,97	33,71	63,33	50,64	68,20	288,58	51,08	302
10	1898,58	60,49	1293,17	58,07	82,63	16,56	118,65	107,11	58,15	83,78	29,35	48,65	316,63	29,39	322

O Gráfico 2 (Figura 20) mostra que PF nem sempre vai contribuir para o aumento da FO, pois à medida que temos muitos quadros B dentro de um GOP

mais compressão sofrerá um vídeo o que leva a uma perda na qualidade da imagem.

Os Gráficos 3 e 4 da Figura 20 mostram que BR e FR tendem a cair à medida que PF aumenta, isso ocorre devido a alta taxa de compressão que um vídeo sofre à medida que temos um aumento de PF.

Em resumo, os experimentos demonstrados neste capítulo comprovam que as variáveis de decisão do modelo proposto se comportam de acordo com as normas e experimentos encontrados na literatura. O modelo proposto procura maximizar a FO para chegar a um equilíbrio entre a qualidade da imagem e a compressão do vídeo através da simulação de um CODEC H.264/AVC.

## **9 EXPERIMENTOS UTILIZANDO O CODEC H.264**

Os experimentos no CODEC analisam o comportamento da busca, pelo algoritmo híbrido proposto, dentro do espaço de soluções, além de analisar e testar diretamente no CODEC H.264 a qualidade das soluções obtidas pelo algoritmo híbrido, SMC.

Os experimentos foram feitos em duas etapas. Na primeira etapa, o SMC foi executado a fim de encontrar soluções de boa qualidade para vídeos do tipo LD, SD e HD. Estas soluções foram então aplicadas ao configurador do CODEC H.264 do projeto DigConv (UNISINOS, 2008a) para que o vídeo resultante desta codificação fosse analisado em termos de compressão e de PSNR. Na segunda etapa foi analisada a influencia do tamanho da Lista Tabu e do nbmax no módulo da Busca tabu do algoritmo híbrido usando-se para isso um vídeo de teste do tipo HD.

### **9.1 Primeira Etapa de Experimentos**

Na primeira etapa dos experimentos foi executado o algoritmo SMC para cada um dos vídeos apresentados na Tabela 16. As melhores soluções encontradas pelo SMC foram usadas para configurar o CODEC H.264 do projeto DigConv (UNISINOS, 2008a). Os vídeos foram codificados pelo CODEC H.264 previamente configurado e como resultado desta codificação obteve-se o PSNR e o grau de compressão

desses vídeos. Esses resultados foram analisados e comparados com os resultados obtidos pela solução inicial.

A Tabela 16 relaciona os tipos de vídeos utilizados no experimento. Na primeira coluna tem-se a identificação do vídeo, ou seja, como ele será referenciado daqui por diante; na segunda coluna tem-se o formato e o nome do vídeo, onde o formato se refere ao formato final que esse vídeo será codificado pelo CODEC H.264, lembrando que todo vídeo que entra no CODEC deve estar obrigatoriamente no formato YUV 4:2:0 (ver seção 2.2); na terceira coluna tem-se a resolução do vídeo; na quarta coluna temos o perfil (descrito na seção 3.2) e o nível (descrito na seção 3.3) do H.264 nos quais o vídeo foi codificado, na quinta coluna tem-se a quantidade de *frames* codificados e na sexta coluna tem-se o tamanho original do vídeo dado em *Kbytes* quando ele está no seu formato original YUV 4:2:0.

Tabela 16 – Vídeos da 1ª fase dos experimentos.

Video	Formato	Resolução	Perfil / Nível	Nro Frames	Tamanho Original (KB)
V02	720p5994 HD - Parkrun	1280 x 720	High/ 40	253	342.688
V10	CIF - Bus	352 x 288	High/ 40	150	22.275
V15	CIF - News	352 x 288	High/ 12	300	44.550
V29	720p HD - Parkrun	1280 x 720	High/ 40	10	342.784
V33	CIF - Bridge close	352 x 288	Main/ 20	50	297.008
V34	SD525 - Pedestrian	720 x 480	Main/ 30	100	50.625

Para esse experimento a equipe DigConv codificou os vídeos da Tabela 16 utilizando valores padrões de configuração, que são valores sugeridos pela norma que rege o padrão H.264 (ITU-T, 2007). Os valores padrões de configuração deram origem à Solução Inicial que foi fornecida ao SMC. Após a codificação dos vídeos da Tabela 16, obteve-se o PSNR e o grau de compressão desses vídeos.

A seguir, os parâmetros BR, FR, QI, QP, QB e PF do CODEC H.264, foram configurados de acordo com os valores encontrados pelo SMC, que na realidade são os valores que compõem a melhor solução encontrada, ou seja, a melhor FO. Novamente os vídeos da Tabela 16 foram codificados pelo CODEC H.264, agora se utilizando parâmetros de configuração obtidos pelo SMC e novamente obtivemos novos valores de PSNR e grau de compressão de vídeo após essa codificação. O PSNR, que é uma métrica objetiva que mede o grau de ruído do vídeo, assume valores dentro do intervalo [0, 50]. PSNR com valores abaixo de 30 dB produz uma qualidade de imagem considerada inaceitável. Neste experimento todos os vídeos tiveram PSNR acima de 30.

Em resumo, um mesmo vídeo foi codificado pelo CODEC H.264 em dois momentos distintos, no primeiro momento a equipe DigConv codificou o vídeo usando valores padrões para o conjunto de parâmetros de configuração do CODEC H.264. Estes parâmetros são os que fazem parte da Solução Inicial. No segundo momento, o vídeo foi codificado usando o conjunto de parâmetros obtidos pela melhor FO encontrada pelo SMC. O PSNR e o grau de compressão obtidos a partir dessas duas codificações foram então comparados e analisados.

### 9.1.1 Resultados do Primeiro Experimento

A Tabela 17 mostra os resultados obtidos após a codificação dos vídeos, relacionados na Tabela 16, que foram codificados pelo CODEC H.264. Cada linha da tabela corresponde aos valores dos parâmetros de configuração do CODEC H.264 usados na codificação de um vídeo, bem como os resultados obtidos em termos de PSNR e grau de compressão desse vídeo. A primeira solução, apresentada para cada vídeo da tabela e que se encontra em destaque, à Solução Inicial proveniente da codificação desse vídeo pela equipe DigConv. Os resultados obtidos por solução inicial, que são o PSNR e grau de compressão do vídeo, serão comparados aos resultados das soluções encontradas pelo SMC. As demais linhas da tabela, que estão agrupadas em um determinado vídeo, correspondem aos parâmetros de configuração do CODEC e os resultados da codificação do vídeo obtidos através da melhor solução encontrada SMC. Em alguns casos foram testadas mais de uma solução do SMC para um mesmo vídeo. Todas as soluções encontradas pelo SMC foram obtidas, após 100 rodadas completas do algoritmo. Para cada rodada completa do SMC, foram configurados o nbmax e a Lista Tabu de forma distinta para se avaliar as diversas soluções encontradas e tentar alcançar uma melhor solução para a parametrização do CODEC e conseqüentemente uma melhor codificação do vídeo.

A coluna 1 identifica o vídeo codificado, que neste caso são os mesmos descritos na Tabela 16. A coluna 2 mostra o valor do nbmax e da Lista Tabu usados pelo SMC para encontrar a melhor solução para se configurar os parâmetros do codificador H.264 para aquele tipo de vídeo específico. A coluna 3 mostra o *bit rate* (BR), em Kbytes/segundo, que o codificador alcançou durante o processo de

codificação do vídeo pelo CODEC H.264. A coluna 4 mostra o *frame rate* (FR), dado em *frames/segundo*.

Tabela 17 – Vídeos codificados no padrão H.264.

Video	nbmax / Lista Tabu	BR (Kbps)	FR (fps)	QI	QP	QB	PF	Média PSNR I	Média PSNR N	FO	Tamanho video (KB)	% PSNR N	% compressão
V02	Solução Ini.	54.400,25	30,00	28	28	30	1	43,383	37,042	380.956,4500	19.694		
	100 / 50	62.499,96	67,46	30	28	30	1	43,319	36,937	437.691,8095	19.511	-0,28%	0,93%
	400 / 100	62.499,99	68,06	27	26	28	1	43,470	37,790	437.692,8355	26.465	2,02%	-34,38%
	600 / 100	62.499,98	67,89	28	30	35	1	43,380	36,340	437.692,3239	12.951	-1,90%	34,24%
V10	Solução Ini.	1.600,35	30,00	28	28	0	0	39,109	39,589	11.234,4500	977		
	50 / 7	19.999,7900	99,12	26	26	21	1	40,298	41,484	140.222,8071	1.764	4,79%	-80,55%
	200 / 100	62.488,3203	99,12	26	26	21	1	40,298	41,484	437.642,5193	1.542	4,79%	-57,83%
	400 / 100	62.499,9922	99,87	26	25	21	1	40,298	41,097	437.725,0055	793	3,81%	18,83%
	800 / 100	62.499,9883	99,45	27	25	29	1	40,298	40,546	437.724,2344	937	2,42%	4,09%
V15	Solução Ini.	246,31	15,00	28	28	30	1	37,357	43,571	1.863,8700	600		
	100 / 60	479,9928	15,16	26	25	27	1	37,357	43,480	3.500,077	594	-0,21%	1,00%
	100 / 50	479,9954	15,06	28	28	30	1	37,357	43,554	3.499,732	598	-0,04%	0,33%
V29	Solução Ini.	25.469,40	60,00	28	28	30	2	43,439	35,183	178.409,500	1.451		
	100 / 50	62.499,9375	67,93	29	28	30	1	43,439	35,222	437.692,150	869	0,11%	40,11%
V33	Solução Ini.	1.995,62	30,00	28	28	30	1	43,036	39,763	14.124,040	804		
	100 / 50	3.999,9575	29,92	29	29	30	1	39,639	39,037	28.154,254	607	-1,83%	24,50%
V34	Solução Ini.	10.036,89	30,00	28	28	30	1	43,101	45,601	70.412,9300	4.044		
	100 / 50	13.999,9141	29,98	29	30	31	1	43,101	45,609	98.153,960	4.057	0,02%	-0,32%

As colunas 5, 6 e 7 mostram respectivamente os valores dos parâmetros de quantização de quadros tipo I (QI), quadros tipo P (QP) e quadros tipo B (QB). A coluna 8 mostra o valor do parâmetro PF que corresponde a quantidade de quadros tipo B dentro do GOP (PF). As colunas de 9 a 14 mostram os resultados obtidos, onde a coluna 9 mostra a média do PSNR do primeiro quadro codificado. A coluna 10 mostra a média do PSNR dos demais quadros codificados, vale ressaltar que cada vídeo de teste tem um número de *frames* a ser codificado e que está definido na Tabela 16. A coluna 11 mostra a Função Objetivo de maior valor encontrado pelo SMC, vale ressaltar que o valor da FO é muito sensível ao valor de suas casas decimais, pois os parâmetros BR e FR que fazem parte da FO, são valores reais positivos e causam um efeito considerável na configuração de um CODEC apenas alterando-se suas casas decimais. A coluna 12 mostra o tamanho do vídeo em *Kbytes* após sua codificação. A coluna 13 mostra o percentual de PSNR ganho (valor positivo) ou perdido (valor negativo) que um vídeo alcançou em relação ao PSNR da solução inicial. Por fim, a coluna 14 mostra o percentual do ganho de compressão (valor positivo) ou perda de compressão (valor negativo) que o vídeo alcançou em relação ao tamanho do vídeo da solução inicial.

De acordo com a Tabela 17, o vídeo V02 foi codificado usando-se três soluções distintas encontradas pelo SMC. Todas as soluções encontradas foram melhores que a Solução Inicial.

A primeira solução encontrada pelo SMC, para o vídeo V02, utilizou um nbmax igual a 100 e uma Lista Tabu igual a 50. Como resultado, obteve-se uma FO melhor que a FO da Solução Inicial. Isso ocorreu porque o SMC conseguiu um ganho de compressão de vídeo de 0,93% ao custo de uma perda de PSNR de apenas 0,28%. Observou-se que o SMC encontrou uma solução que gerou uma pequena perda na qualidade da imagem, refletida no valor do PSNR, porém essa perda não levou a uma qualidade de imagem inaceitável, que são valores de PSNR menores que 30, portanto, além deste fato não ter ocorrido, o vídeo alcançou um maior grau de compressão.

A segunda solução para o vídeo V02 foi obtida pelo SMC usando um nbmax igual a 400 e uma Lista Tabu igual a 100. Esta solução obteve um ganho de 2,02% no PSNR, porém o tamanho do vídeo aumentou em 34,38%. Observa-se que neste caso o SMC considerou esta FO melhor que a FO da primeira solução devido ao fato de ele priorizar, em sua FO, o ganho de PSNR.

A terceira solução para V02 foi obtida pelo SMC usando um nbmax igual a 600 e uma Lista Tabu igual a 100. Esta solução não alcançou um ganho no PSNR, apenas conseguiu minimizar as perdas dessa métrica, tal como ocorreu na primeira solução, por outro lado, obteve-se um ganho de compressão de vídeo de 34,24%. Tal qual ocorreu na segunda solução de V02, a perda de PSNR não levou a um valor de PSNR inaceitável.

Nos experimentos feitos com o vídeo V02 pôde-se observar que o comportamento do SMC foi de priorizar o ganho do PSNR. Observou-se que o maior valor da FO foi obtido quando se conseguiu ter um ganho de PSNR, por outro lado, quando esse ganho não é alcançado, o SMC tende a minimizar ao máximo a perda do PSNR e então dá prioridade ao ganho de compressão de vídeo, como uma forma de compensar a perda da qualidade da imagem. Nesse primeiro momento pode-se ter a impressão de que o SMC tende a perder qualidade de imagem para ganhar compressão, o que parece meio óbvio, porém isso é uma conclusão prematura até vermos os resultados dos demais vídeos.

De acordo com a Tabela 17, o vídeo V10 foi codificado usando-se quatro soluções distintas encontradas pelo SMC. Todas as soluções encontradas foram

melhores que a solução inicial. A primeira solução encontrada pelo SMC utilizou um nbmax igual a 50 e uma Lista Tabu igual a 7. Esta solução obteve um ganho de 4,79% no PSNR e uma perda de 80,55% na compressão. Esta solução obteve o pior valor da FO em relação às outras soluções encontradas pelo SMC para este mesmo vídeo, porém com nbmax e tamanho Lista Tabu maiores.

A segunda solução para V10 foi obtida pelo SMC usando um nbmax igual a 200 e Lista Tabu igual a 100. Esta solução obteve um ganho de 4,79% no PSNR e uma perda de 57,83% na compressão. Se compararmos as FOs da primeira solução com a da segunda solução verifica-se que as duas obtiveram o mesmo percentual de ganho no PSNR, porém com perdas de compressão distintas. O SMC considerou como melhor FO aquela que obteve menos perdas na compressão, provando que o SMC considera em primeiro plano o ganho do PSNR e em segundo plano tenta minimizar as perdas de compressão.

A terceira solução para V10 foi obtida pelo SMC usando um nbmax igual a 400 e uma Lista Tabu igual 100. Esta solução obteve um ganho de 3,81% no PSNR e um ganho de 18,83% na compressão. Esta solução obteve o melhor valor da FO dentre todas as outras. Nota-se que os resultados obtidos por esta solução foram de ganhos tanto no PSNR quanto na compressão, tal qual ocorreu na quarta solução, porém os ganhos obtidos nesta terceira solução foram maiores que os obtidos na quarta solução. Com esse resultado observamos que o principal comportamento do SMC se revelou, ou seja, a melhor solução é aquela que não tem nenhum tipo de perdas.

A quarta solução para V10 foi obtida pelo SMC usando um nbmax igual 800 e uma Lista Tabu igual 100. Esta solução obteve um ganho de 2,42% no PSNR e um ganho de 4,09% na compressão. Este foi o segundo maior valor da FO encontrado pelo SMC para o vídeo V10. Essa solução conseguiu obter ganhos tanto na qualidade da imagem quanto na compressão do vídeo.

Em resumo, as diversas soluções usadas para se codificar o vídeo V10, mostram de forma mais completa os diversos comportamentos do SMC, onde deixa claro que as melhores soluções são aquelas que alcançam os maiores ganhos tanto de PSNR quanto de compressão enquanto que as piores soluções foram aquelas em que se obteve algum tipo de perda. Isso comprova que o SMC busca maximizar a qualidade da imagem e o grau de compressão e quando isso não ocorre, o SMC tende a equilibrar esses dois fatores e o valor da FO diminui.



De acordo com a Tabela 17, o vídeo V15 foi codificado usando-se duas soluções distintas encontradas pelo SMC. Todas as soluções encontradas foram melhores que a solução inicial. A primeira solução utilizou um nbmax igual a 100 e uma Lista Tabu igual a 60. Esta solução obteve uma perda de 0,21% no PSNR e um ganho de compressão de 1%. Observou-se que o SMC apesar de não conseguir um ganho no PSNR, alcançou um maior grau de compressão do vídeo e tentou minimizar ao máximo a perda do valor do PSNR.

A segunda solução V15 foi obtida pelo SMC usando um nbmax igual a 100 e uma Lista Tabu igual a 50. Esta solução obteve uma perda de 0,04% no valor do PSNR, que foi uma perda menor em relação à primeira solução de V15, e um ganho de 0,33% de compressão. Observou-se que a primeira solução obteve um melhor valor da FO em relação ao valor da FO da segunda solução. A análise dessas duas FOs mostrou que quando o SMC não alcança um ganho no PSNR, ele tenta maximizar a compressão. Esse comportamento é o mesmo encontrado quando se analisa a primeira e segunda solução do vídeo V02, ou seja, as duas FOs obtiveram perda de PSNR porém o melhor valor da FO foi aquela que obteve maior compressão.

De acordo com a Tabela 17, os vídeos V29, V33 e V34 foram codificados usando-se soluções que foram encontradas pelo SMC e este utilizou nbmax igual 100 e tamanho da Lista Tabu igual 50. O resultado da codificação do V29 obteve ganhos de 0,11% no PSNR e de 40,11% na compressão, o que é um ótimo resultado, pois não houve perdas, nem de qualidade de imagem nem de compressão. O resultado da codificação do V33 obteve uma perda de 1,83% no PSNR mas em compensação alcançou um ganho de compressão de 24,50%. O resultado da codificação do V34 obteve um pequeno ganho de 0,02% no PSNR e uma pequena perda de compressão de 0,32%.

A Figura 21 mostra graficamente a evolução dos valores das Funções Objetivo que foram obtidas para o vídeo V02 (conforme Tabela 17), usando-se para isso, diferentes configurações de valores de nbmax e tamanho da Lista Tabu para algoritmo híbrido SMC. Observou-se que o melhor valor da FO foi alcançado quando o SMC utilizou um nbmax igual a 400 e um tamanho de Lista Tabu igual a 100.

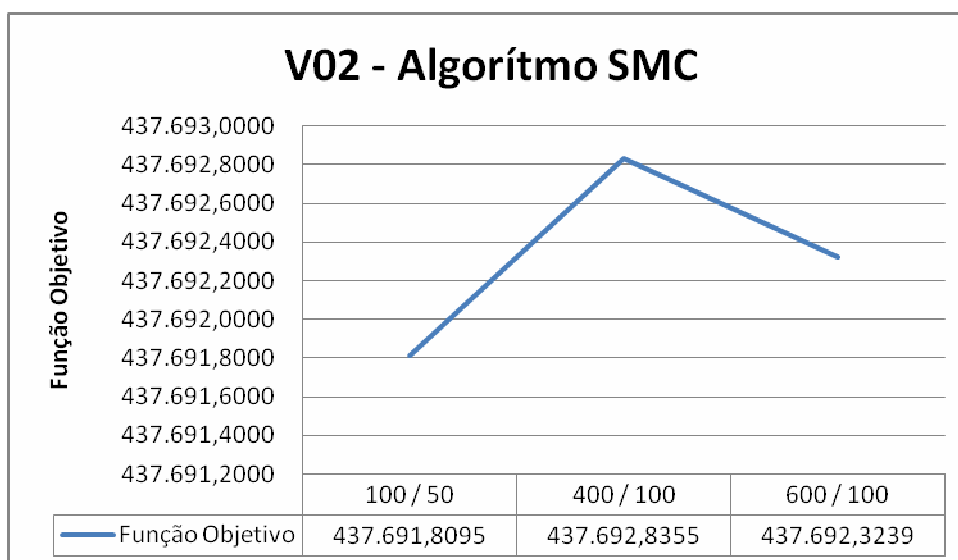


Figura 21 – Algoritmo SMC aplicado ao Vídeo V02.

A Figura 22 mostra os resultados alcançados em termos de PSNR e de compressão de vídeo para o vídeo V02 (conforme Tabela 17). Esses resultados foram obtidos quando se configurou o CODEC H.264 de acordo com as soluções encontradas pelo SMC. Observando-se o gráfico da Figura 22 temos na linha horizontal as soluções 1, 2 e 3 que correspondem às FOs apresentadas no gráfico da Figura 21.

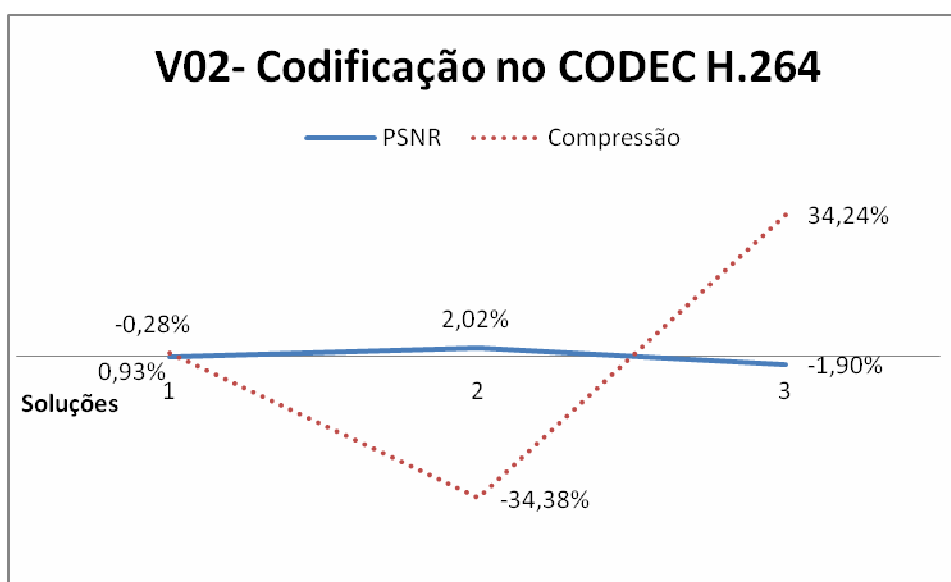


Figura 22 – Resultados da codificação do vídeo V02 quanto ao PSNR e a compressão.

Comparando-se os gráficos dessas das figuras 20 e 21 observa-se que o melhor valor da FO mostrado na Figura 21 corresponde à solução 2 na Figura 22. Na solução 2, encontramos o mais alto ganho de PSNR e uma perda de

compressão significativa. Isso ocorreu porque o SMC priorizou o ganho de qualidade de imagem que se reflete no aumento do PSNR.

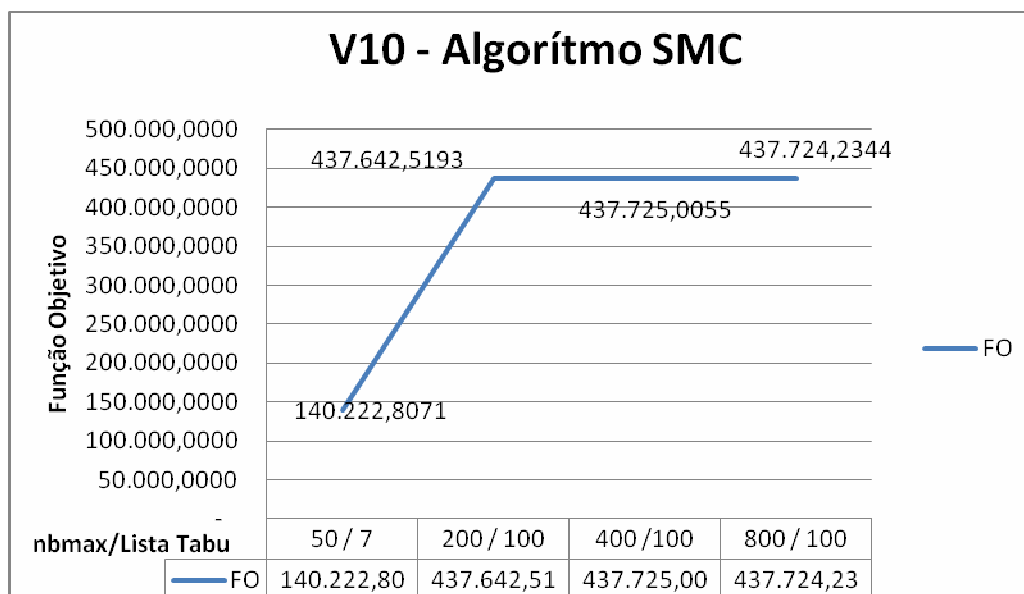


Figura 23 – Algoritmo SMC aplicado ao Vídeo V10.

A Figura 23 mostra a evolução do valor da FO do vídeo V10 (conforme Tabela 17) para as diversas configurações de valores do nbmax e tamanho da Lista Tabu do algoritmo híbrido SMC. Pode-se observar que a melhor valor da FO foi obtido quando o SMC utilizou um nbmax igual a 400 e um tamanho de Lista Tabu igual a 100.

Observou-se que os vídeos V02 e V10 obtiveram seus melhores valores da FO quando o SMC usou um nbmax igual a 400 e tamanho da Lista Tabu igual 100., o que leva a crer que essa é uma boa configuração para a Busca Tabu do algoritmo híbrido SMC.

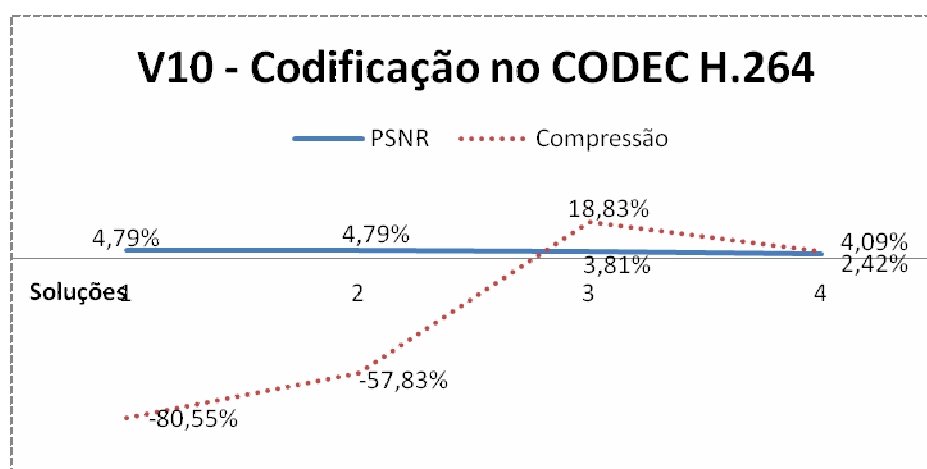


Figura 24 – Resultados da codificação de V10 em termos de PSNR e Compressão.

A Figura 24 mostra os resultados alcançados em termos de PSNR e de compressão para o vídeo V10 (conforme Tabela 17). Esses resultados foram obtidos quando o CODEC H.264 foi configurado de acordo com as soluções encontradas pelo SMC. Observando o gráfico temos na linha horizontal as soluções 1, 2, 3 e 4 que correspondem às FOs apresentadas na Figura 23.

Na Figura 24 fica claro que a dinâmica do SMC busca dar prioridade ao ganho de PSNR e em seguida ao ganho de compressão. Nota-se que as soluções 1 e 2 obtiveram o mesmo ganho de PSNR, porém para o SMC o melhor valor da FO entre as duas soluções foi aquela em que se conseguiu minimizar a perda de compressão, que é o caso da solução 2. Outro comportamento interessante, que pode ser observado na Figura 24, é que as soluções 3 e 4 alcançaram ganhos de PSNR e de compressão, porém o melhor valor da FO foi aquele cujos ganhos foram os maiores, que é o caso da solução 4.

Neste capítulo mostrou-se os resultados obtidos na codificação dos vídeos de teste pelo CODEC H.264. Ele foi configurado usando-se as diversas soluções obtidas nas diversas rodadas do Algoritmo Híbrido SMC. Procurou-se diversificar cada rodada do SMC quanto a diferentes configurações dos parâmetros nbmax e tamanho da Lista Tabu usados pelo módulo da Busca Tabu desse algoritmo. Os vídeos V02 e V10 possuem formatos de codificação bastante distintos e por isso foram os escolhidos para os testes de codificação usando diferentes configurações de parâmetros do CODEC H.264. Sabendo-se que essas diversas configurações de parâmetros para o CODEC H.264 vieram das melhores soluções encontradas pelo algoritmo híbrido SMC. O vídeo V02 foi codificado no formato HD e o V10 foi codificado no formato CIF.

Constatou-se em todos os experimentos que o comportamento do SMC independe do formato do vídeo. Os maiores ganhos de compressão foram obtidos quando se codificou um vídeo no formato HD, que é o caso dos vídeos V02 e V29 mostrados na Tabela 17. As soluções encontradas, para os vídeos V02 e V29, pelo SMC conseguiram um ganho de 34,24% e 40,11% respectivamente no grau de compressão desses vídeos. O tempo médio de uma execução completa do SMC foi de 4 minutos, sabendo-se que uma rodada completa equivale a 100 execuções do SMC.

A Figura 25 compara os tamanhos compactados dos vídeos obtidos, através da solução inicial, com os tamanhos obtidos pela melhor valor da FO do SMC. As

soluções apresentadas nesta figura são as melhores FOs obtidas para cada um dos vídeo da Tabela 17.

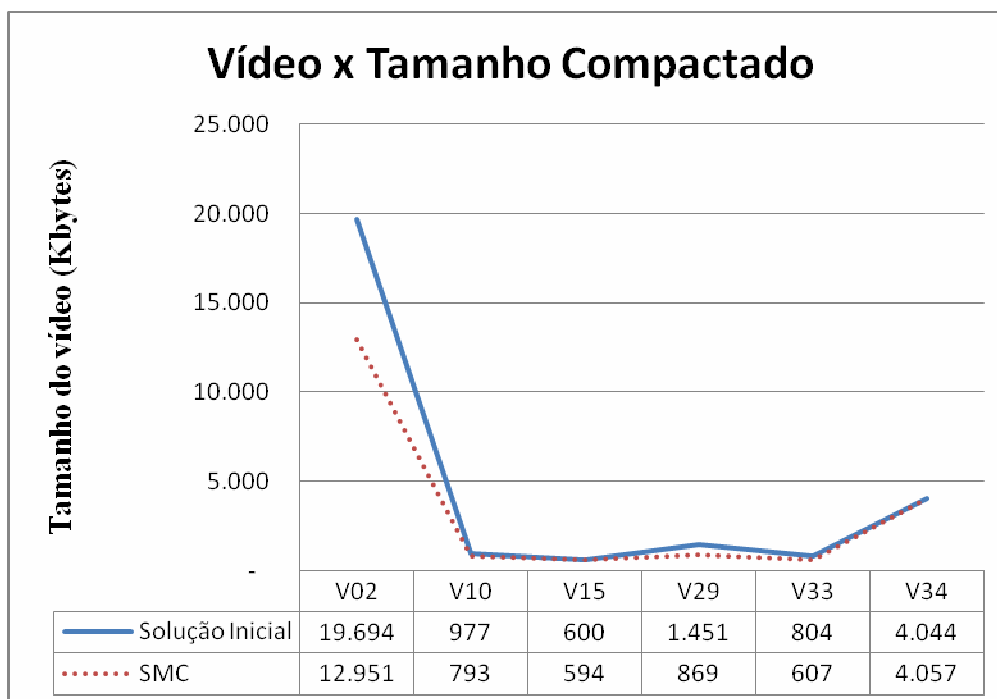


Figura 25 – Comparação entre os tamanhos finais dos vídeos codificados pelo CODCE H.264 a partir da Solução Inicial e a da Solução do SMC.

A Figura 25 mostra que a tendência do SMC é buscar uma maior compactação do vídeo. No gráfico pode-se observar que todos os vídeos tiveram seu tamanho compactado diminuído em relação ao tamanho compactado da solução inicial, com exceção do vídeo V34 que ficou 0,32% acima do tamanho obtido pela solução inicial, porém o SMC conseguiu um ganho de PSNR de 0,02%.

A Figura 26 mostra graficamente uma outra tendência da dinâmica do SMC, que é de buscar uma melhor qualidade de imagem através do aumento do PSNR. Nesta Figura pode-se observar os PSNRs alcançados pelos melhores valores das FOs de todos os vídeos que se encontram na Tabela 17.

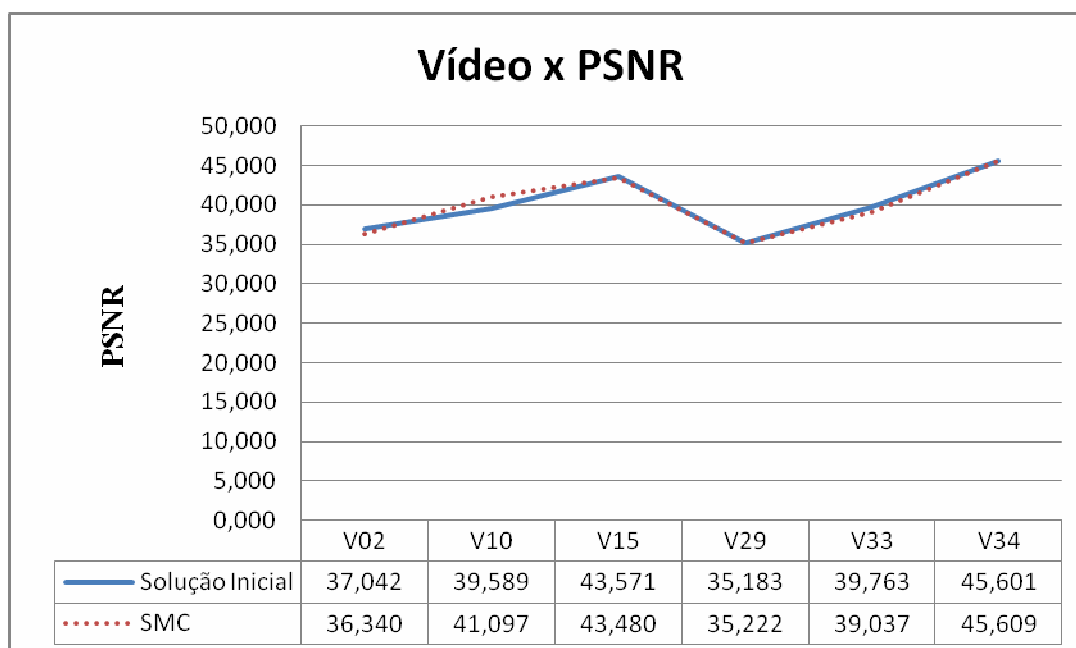


Figura 26 – PSNRs da Solução Inicial e do SMC.

Pode-se observar que quando o SMC não conseguiu obter um PSNR melhor que o da solução inicial, ele se manteve próximo a ele. Observou-se que os vídeos que alcançaram um PSNR melhor que o da Solução Inicial foram os vídeos V10, V29 e V34 e que correspondem a 50% dos vídeos testados. De acordo com Tabela 16 esses vídeos são de formatos distintos, o primeiro possui formato CIF, o segundo é de formato HD e o terceiro é de formato SD. Com isso observou-se que o SMC não é tendencioso a melhorar a qualidade da imagem para um determinado formato de vídeo e sim para os diversos formatos testados.

Tabela 18 – Médias e desvios-padrão dos algoritmos BT, AG e Híbrido.

Video	Nro Execuções	Busca Tabu			Algoritmo Genético			Algoritmo Híbrido	
		nbMax/ Lista Tabu	Média da FO	Desvio Padrão	Ger. / TxCros. / TxMut.	Média da FO	Desvio Padrão	Média da FO	Desvio Padrão
V02	100	100 / 50	437.639,39	41,97	100 / 80 / 20	437.652,47	29,44	437.664,25	19,62
V10	100	200 / 100	437.682,28	34,75	100 / 80 / 20	437.694,16	23,11	437.704,45	17,01
V15	100	100 / 50	3.497,44	1,85	100 / 80 / 20	3.497,96	1,44	3.498,42	0,92
V16	100	100 / 50	5.406,04	1,32	100 / 80 / 20	5.406,43	1,00	5.406,69	0,76
V29	100	100 / 50	437.641,53	39,13	100 / 80 / 20	437.653,76	26,99	437.664,66	18,66
V30	100	100 / 50	912,26	0,39	100 / 80 / 20	912,44	0,26	912,54	0,19
V33	100	100 / 50	28.143,62	10,03	100 / 80 / 20	28.145,94	7,28	28.148,08	5,33
V34	100	100 / 50	98.134,63	17,26	100 / 80 / 20	98.138,82	11,27	98.143,69	6,80

A Tabela 18 apresenta a média dos valores da Função Objetivo obtidos pelos módulos da Busca Tabu, do Algoritmo Genético e do Algoritmo Híbrido (SMC) e seus respectivos desvios-padrão. A coluna 1 da tabela identifica o vídeo (conforme Tabela 16); a coluna 2 identifica o número de execuções completas que cada

algoritmo foi submetido; a coluna 3 identifica o nbmax e o tamanho da Lista Tabu utilizados na Busca Tabu e no algoritmo Híbrido; a coluna 4 mostra o valor médio dos melhores valores das Funções Objetivo encontrados pela Busca Tabu seguida pelos seus respectivos desvios-padrão na coluna 5; a coluna 6 mostra o número de gerações, a taxa de *crossover* e a taxa de mutação utilizados pelo Algoritmo Genético e pelo Algoritmo Híbrido; a coluna 7 mostra o valor médio das melhores Funções Objetivo encontradas pelo Algoritmo Genético seguidos pelos seus respectivos desvios-padrão apresentados na coluna 8; a coluna 9 mostra o valor médio das melhores Funções Objetivo encontradas pelo Algoritmo Híbrido seguidos pelos seu respectivos desvios-padrão apresentados na coluna 10.

Observou-se na Tabela 18 que o Algoritmo Híbrido obteve melhores soluções para todos os vídeos usados no experimento. Ele também obteve o menor desvio-padrão em comparação com a Busca Tabu e o Algoritmo Genético. Atualmente as pesquisas e trabalhos feitos com metaheurísticas estão se voltando para o uso de técnicas híbridas (Burke *et al*, 1998, 2001; Gómez *et al*, 2005) porque elas tem demonstrado melhores resultados.

De acordo com os desvios-padrão dos três algoritmos, podemos notar que a Busca Tabu operou num espaço de busca de maior diversificação, enquanto que o Algoritmo genético assumiu um papel de intensificador da solução, pois seu desvio-padrão foi sempre menor que o da Busca Tabu. A diversificação da Busca Tabu ocorreu devido ao fato da mesma não estar usando políticas de intensificação, além do que, sua geração de vizinhança foi feita através da perturbação de todas as variáveis de decisão da FO, o que contribui ainda mais para uma maior diversificação.

O Algoritmo Genético convergiu para uma política de intensificação dentro do algoritmo híbrido, devido ao fato deste usar o *crossover* aritmético. Segundo (Michalewicz, 1996), sistemas que fazem uso de *crossover* aritmético são mais estáveis e produzem melhores soluções com desvios-padrão muito menores.

## 9.2 Segunda Etapa de Experimentos

A segunda etapa de experimentos consistiu em analisar o processo de exploração do espaço de soluções realizado pelo módulo da Busca Tabu do

algoritmo híbrido SMC. Para isso foram avaliados os parâmetros: tamanho da Lista Tabu e nbmax. O tamanho da Lista Tabu tem impacto no processo de intensificação do algoritmo enquanto que o nbmax tem impacto sobre o tempo de busca e é diretamente proporcional ao tempo computacional.

Nesta etapa o vídeo analisado não foi submetido ao CODEC H.264 como nos experimentos anteriores, apenas foi analisado o valor da FO encontrado em função do nbmax e do tamanho da Lista Tabu.

Os experimentos foram realizados com o vídeo V02, que é do tipo HD, e como já foi citado, este tipo de vídeo é o que consegue alcançar maiores ganhos de compressão nas soluções alcançadas pelo SMC.

### 9.2.1 Resultados do Segundo Experimento

A Tabela 19 mostra os resultados obtidos para o vídeo V02 quando se aplicou uma Busca Tabu com variação do parâmetro nbmax. Neste experimento, enquanto nbmax variava, o tamanho da Lista Tabu permaneceu igual a 100. Buscando-se uma solução não tendenciosa, cada solução foi obtida a partir de 100 execuções do algoritmo.

Tabela 19 –BT com variação do nbmax aplicada ao vídeo V02.

nbmax	Lista Tabu	FO
100	100	437.692,691
200	100	437.692,424
300	100	437.692,611
400	100	437.692,835
500	100	437.692,770
600	100	437.692,324
700	100	437.692,552
800	100	437.692,362
900	100	437.692,672
1000	100	437.692,790
2000	100	437.692,874

De acordo com a Tabela 19 a melhor Função Objetivo foi obtida com o nbmax igual a 2000. O nbmax permite que a busca se prolongue nas regiões promissoras, por isso, quanto maior o nbmax maior a chance de se encontrar melhores soluções. Vale ressaltar que o tempo computacional eleva-se com o aumento do nbmax. Neste



experimento o tempo de execução para o nbmax igual a 100 foi de 13 minutos enquanto que para o nbmax igual a 1000 foi de 46 minutos e 47 segundos e para o nbmax igual a 2000 foi de 1 hora e 5 minutos.

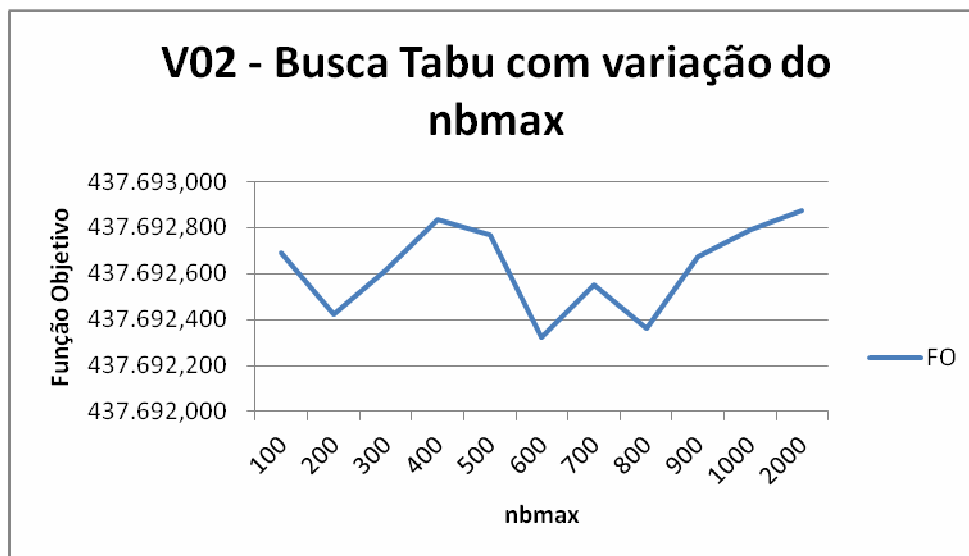


Figura 27 – BT com variação do nbmax aplicada ao vídeo V02.

A Figura 27 mostra a evolução da Função Objetivo ao longo do aumento do nbmax dentro da Busca Tabu. Pode-se verificar que o aumento do nbmax promove melhorias nas soluções encontradas. Nota-se que ao final do gráfico a tendência é que quanto maior o nbmax melhor tende a ser a solução encontrada.

Tabela 20 – BT com variação do tamanho da lista tabu aplicada ao V02.

nbmax	Lista Tabu	FO
100	40	437.692,06149
100	50	437.691,80947
100	60	437.690,58016
100	70	437.691,82712
100	100	437.691,09380
100	150	437.691,79710
100	200	437.691,91119
100	300	437.692,48785
100	400	437.692,40009
100	500	437.692,67496
100	600	437.692,32395
100	700	437.692,15122
100	800	437.692,48982
100	900	437.691,97956
100	1000	437.691,52058

A Tabela 20 mostra os resultados obtidos para o vídeo V02 quando se aplica uma Busca Tabu com variação do tamanho da Lista Tabu. Enquanto o tamanho da

Lista Tabu varia, o nbmax permanece igual a 100. Buscando-se uma solução não tendenciosa, cada solução foi obtida a partir de 100 execuções do algoritmo.

A Lista Tabu tem a característica de diversificar a busca, guiando o algoritmo para regiões ainda não exploradas no espaço de soluções. Desse modo, com o aumento desse valor, a Busca Tabu acaba diversificando a busca, impedindo que regiões promissoras sejam exploradas mais intensamente.

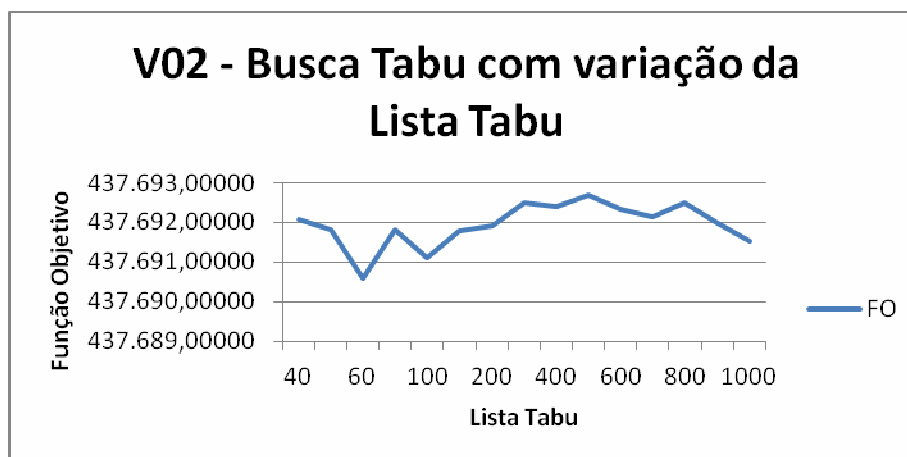


Figura 28 – BT com variação do tamanho da Lista Tabu aplicada a V02.

Analisando os experimentos com o tamanho da Lista Tabu, verifica-se que a melhor solução foi obtida com o tamanho da Lista Tabu igual a 500. Também se verificou que o tempo de execução para uma Lista Tabu de tamanho igual a 60 foi de 4 minutos e 34 segundos e para um tamanho igual a 1000 foi de 5 minutos e 25 segundos, ou seja, o aumento do tamanho deste parâmetro não implica em um tempo computacional muito maior, diferente do que ocorre com o aumento do nbmax.

A Figura 28 mostra a variação da Função Objetivo para o vídeo V02 quando a Busca Tabu tem variações no tamanho da Lista Tabu. Nesta figura se percebe claramente que o aumento exagerado do tamanho da Lista tabu não necessariamente vai encontrar melhores soluções. A Lista Tabu permite que a busca escape de ótimos locais e seja guiada para outras áreas do espaço de soluções. Contudo, um valor muito elevado impede que as regiões promissoras do espaço de soluções sejam devidamente exploradas.

Ao se fazer uma análise geral desses experimentos, verifica-se que deve-se ter bom senso ao se parametrizar a Busca Tabu. O aumento exagerado do tamanho

da Lista Tabu acaba guiando a busca para regiões pouco promissoras do espaço de soluções. Por outro lado, o aumento do *nbmax*, permite que a Busca Tabu prolongue seu processo de exploração, muitas vezes, consumindo tempo computacional sem prover nenhum benefício. Isto significa que equívocos na parametrização do algoritmo pode gerar uma busca superficial abrangendo um grande espaço de soluções sem que seja feita uma busca aprofundada em regiões promissoras.

## 10 CONCLUSÃO

Neste trabalho foi desenvolvido e implementado um algoritmo híbrido denominado Simulador de Metaheurísticas aplicado a um CODEC (SMC) que utiliza a Busca Tabu em combinação com o Algoritmo Genético, com o objetivo de estudar o comportamento de seis dos principais parâmetros de configuração do CODEC de vídeo H.264/AVC. A partir do estudo do comportamento da dinâmica desses parâmetros, validou-se o algoritmo híbrido proposto. Experimentos foram feitos com o SMC visando obter boas soluções para a configuração do CODEC H.264.

Os parâmetros do CODEC H.264 que foram estudados e que formaram o conjunto das variáveis de decisão, foram: o *bit rate*; o *frame rate*; os parâmetros de quantização para quadros tipo B, tipo P e tipo I; e quantidade de quadros tipo B que compõem um Grupo de Imagens (GOP). O algoritmo híbrido proposto busca a melhor configuração desses parâmetros do CODEC H.264 para que este codificador codifique um vídeo com melhor qualidade de imagem e maior compressão.

O SMC foi validado através dos padrões de comportamento dos parâmetros, que foram encontrados na literatura, e se mostrou bastante eficiente segundo os experimentos realizados. Foi observado que o algoritmo SMC assume que a solução de melhor qualidade é aquela que configura o CODEC H.264 de maneira que o vídeo tenha um ganho de qualidade de imagem e ao mesmo tempo um ganho no grau de compressão em relação à solução inicial. Quando esta solução não é alcançada, o SMC tenta chegar a um equilíbrio entre o ganho de qualidade de imagem e a perda de compressão ou vice-versa.

De acordo com os experimentos realizados, o SMC tende a preservar o ganho de qualidade da imagem, ou seja, o aumento do PSNR. Caso o SMC encontre duas soluções que geram um mesmo ganho de qualidade de imagem, então ele optará pela que tiver um maior ganho de compressão, ou se nenhuma das soluções possuírem tal ganho de compressão, então ele optará pela solução que leva a uma menor perda de compressão. Isso ocorre porque nem sempre uma solução alcançará ganho de qualidade de imagem e de compressão ao mesmo tempo.

Outra dinâmica de comportamento do algoritmo observada, é quando o SMC não consegue soluções com ganho de qualidade de imagem, então ele busca a solução que tem maior ganho de compressão, ou seja, quando não há ganho de PSNR ele prioriza o ganho de compressão. Esse comportamento ocorreu nos experimentos realizados com os vídeos V02 e V15.

Foi observado que o máximo de perda de qualidade de imagem gerado pelo SMC foi de 1,90% e o máximo em perda de compressão chegou a 80,55%. Esse comportamento mostra que a perda de qualidade de imagem é minimizada ao máximo pelo SMC.

O algoritmo SMC provou ser robusto, suportando os três formatos de vídeo, LD, SD e HD, que um CODEC H.264 pode codificar, além de também ter se mostrado confiável, pois todas as soluções testadas trouxeram uma melhora na codificação do vídeo. A dinâmica de comportamento do SMC traduz a dinâmica de CODEC real (JIGA de testes). As variáveis de decisão escolhidas, se mostraram representativas, pois viabilizaram uma correta análise em termos de qualidade de imagem e compressão de vídeo.

É nitidamente observado o ganho computacional que esta ferramenta proporciona ao CODEC H.264, pelo fato dela ter o poder de melhorá-lo sem que seja necessário a alteração de seu código-fonte. O simulador proposto consegue maximizar a performance do referido CODEC trabalhando apenas com a configuração adequada de seus parâmetros mais representativos.

Em termos de tempo de execução do SMC, observou-se dentro de um ciclo de 100 execuções, o SMC levou um tempo médio de 4 minutos. Seu código foi

construído na linguagem C ANSI e todos os experimentos foram feitos em uma máquina com processador Intel Dual Core.

### **10.1 Trabalhos Futuros**

Como trabalhos futuros é possível aprimorar o modelo acrescentando outras variáveis decisão que influenciem diretamente na qualidade da imagem, como por exemplo, a quantidade de partições de um macrobloco, a dinâmica de um vídeo, a quantidade de frames de referência a serem considerados.

No que tange ao módulo da Busca Tabu que compõe o algoritmo híbrido, pode-se estudar a utilização de diferentes técnicas de intensificação e diversificação, bem como novas estruturas de vizinhança.

Quanto ao módulo do Algoritmo Genético que compõe o algoritmo híbrido, pode-se fazer experimentos com esse módulo através de alterações em seus parâmetros, como a taxa de mutação, a taxa de cruzamento, o tamanho da população e tipos de *crossover*.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ABNT, Associação Brasileira de Normas Técnicas. NBR 15602-1. Televisão Digital Terrestre – Codificação de vídeo, áudio e multiplexação, Parte 1: Codificação de vídeo. Rio de Janeiro, 2007.
- Alencar, M. S. Televisão Digital. São Paulo, Ed. Érica, 2007.
- Anderson, C. A., Jones, K. F., Parker, M., Ryan, J., Path assignment for call routing: an application of tabu search. *Ann. Ops Res.* 41, 301-312, 1993.
- Barnes, J. W., Laguna, M., Solving the multiple-machine weighted flow time problem using tabu search. *IIE Trans.* 25, 121-130, 1993.
- Beyer, D., Ogier, R., Tabu learning: a neural network search method for solving nonconvex optimization problems. *Prec. Int. Joint Conf. Neural Networks. IEEE and INNS*, Singapore, 1991.
- Bland, J. A., Dawson, G. P., Tabu search and design optimization. *Computer-Aided Des.* 23, 195-202, 1991.
- Bland, J. A., Dawson, G. P., Tabu search and design optimization. *Computer-Aided Des.* 23, 195-202, 1991.
- Bovet, J., Constantin, C., de Werra, D., A convoy scheduling problems. *Discrete Appl. Math.* (in press).
- Burke, E.K., Causmaecker, P.D., Berghe, G.V., *A hybrid Tabu Search Algorithm for the Nurse Rostering Problem*. B. McKay et al (eds.), *Simulated Evolution and Learning*, Springer, Lecture Notes in Artificial Intelligence v.1585, p.187-194, 1998.
- Burke, E.K., Causmaecker, P.D., Berghe, G.V., *A Memetic Approach to the nurse Rostering Problem*. *Applied Intelligence* 15, p.199-214, Kluwer Academic Publishers, 2001.
- Cermak, G., Pinson M. and Wolf, S., The relationship among video quality, screen resolution, and bit rate. *IEEE transactions on broadcasting*, v. 30(2), 2011.
- Chakrapani, J., Skorin-Kapov, J., Massively parallel tabu search for the quadratic assignment problem. *Ann. OpsRes.* 41, 327-342, 1993.
- Correia, P. L., Pereira, F., Objective evaluation of video segmentation. *IEEE Transaction on image processing*, vol. 12, no. 2, 2003.
- CPqD. Arquitetura de Referência. Sistema Brasileiro de Televisão Digital Terrestre. 2006a. Versão AA. PD.30.12.34A.0001A/RT-13/AA. Disponível em <<http://sbtvd.cpqd.com.br/>> Acessado em maio, 2011.
- CPqD. Especificação Técnica de Referência. 2006b. Versão AA. PD.30.12.34A.0001A/RT-14/AA. Disponível em <<http://sbtvd.cpqd.com.br/>> Acessado em maio, 2011

Da Silva, A. M. C., Um estudo sobre o padrão H.264/AVC de compressão de vídeo. 2007. Dissertação (Mestrado em Informática). Universidade Católica de Pelotas, Pelotas, 2007.

Dammeyer, F., Voss, S., Dynamic tabu list management using the reverse elimination method. *Ann. Ops Res.* 41,31-46, 1993.

Daniels, R. L., Mazzola, J. B., A tabu search heuristic for the flexible-resource flow shop scheduling problem. *Ann. Ops Res.* 41, 207-230, 1993.

de Werra, D., Hertz, A., Tabu search technique: a tutorial and an applications to neural networks. *OR Spectrum* 11,131-141, 1989.

Deb, K., *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, New York, 2001.

DelrAmico, M., Trubian, M., Applying tabu search to the job-shop scheduling problem. *Ann. Ops Res.* 41, 231-252, 1993.

Friden, C., Hertz, A., de Werra, D., Stabulus: a technique for finding stable sets in large graphs with tabu search. *Computing* 42, 35-44, 1989.

Garey, M. R., Johnson, D. S., Stockmeyer, L., Some simplified NP-complete problems, *Proceedings of the sixth annual ACM symposium on Theory of computing*, p.47-63, April 30-May 02, 1974, Seattle, Washington, United States.

Gendreau, M., Hertz, A., Laporte, G., A tabu search heuristic for the vehicle routing problem. *Managmt Sci.* (in press).

Gendreau, M., Lori, M., Laporte, G. and Martello, S., A Tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks*, 51:4-18, 2008.

Gendreau, M., Soriano, P., Salvail, L., Solving the maximum clique problem using a tabu search approach. *Ann. Ops Res.* 41, 385-404, 1993.

Glover, F., Laguna, F., *Tabu Search*, Kluwer Academic Publishers, Norwell, MA, 1997

Glover, F., Ejection chains, reference structures, and alternating path methods for the traveling salesman problem. *Graduate School of Business and Administration, University of Colorado at Boulder*, 1992.

Glover, F., Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research* v.1, p.533-549, 1986.

Glover, F., Kelly, J. P., Laguna, M., Genetic algorithms and tabu search: hybrids for optimization. *Computers and Operations Research*, 22 (1), 111± 134, 1995.

Glover, F., Kochenberger, G. A., *Handbook of Metaheuristics*. Kluwer Academic Publishers, 2003.

Glover, F., Laguna, M., Bandwidth packing: a tabu search approach. *Managmt ScL* 39, 492-500, 1993.

Glover, F., Laguna, M., *Modern Heuristic Techniques for Combinatorial Problems*, chapter 3 – Tabu Search, pages 70–150. McGraw Hill, 1993.

Glover, F., McMillan, C., Novick, B. Interactive decision software and computer graphics for architectural and spaceplanning. *Ann. Ops. Res.* 5, 557-573, 1985.

Glover, F., McMillan, C., The general employee scheduling problem: an integration of management science and artificial intelligence. *Computers Ops Res.* 15, 563-593, 1986.

Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, Inc., Reading, MA, 1989.

Golston, J., Rao, A., Video codecs tutorial: Trade-offs with H.264, VC-1 and other advanced codecs. *Embedded Systems Conference Silicon Valley*, 2006.

Gómez, A. T.; Oliveira, L. P. L. de; Severino, P. A. R.; Queiroz, R. B. Desenvolvimento de um Sistema Híbrido para a Classificação de Imagens de Satélites. In: XXXVII Simpósio Brasileiro de Pesquisa Operacional, 2005, Gramado. *Anais do XXXVII Simpósio Brasileiro de Pesquisa Operacional*. Santa Maria: Universidade Federal de Santa Maria, 2005. v. único. p. 259-266.

Gonçalves, José Fernando, Mendes, J. J. M., Resende, Maurício G. C. A hybrid genetic algorithm for the job shop scheduling problem. *European Journal of Operation Research*, 167, 77-95, 2005.

Gonzales, R., Woods, R. *Processamento de Imagens Digitais*. São Paulo: Edgar Blücher, 2003.

Hansen, P., Jaumard, B., Algorithms for the maximum satisfiability problem. *Computing* 44, 279-303, 1990.

Hansen, P., Jaumard, B., Da Silva, Average linkage divisive hierarchical clustering. *J. Classif.* (in press).

Hansen, P., Jaumard, B., Poggi di Aragao, M., Mixed integer column generation algorithms and the probabilistic maximum satisfiability problem. *Prec. 2nd Integer Programming and Combinatorial Optimization Conf.* Carnegie Mellon, 1992.

Hertz, A., de Werra, D., Using tabu search techniques for graph coloring. *Computing* 29, 345-351, 1987.

Hertz, A., Jaumard, B., Poggi di Aragao, M., Topology of local optima for the K-coloring problem. *Discrete Appl. Math.* 49, 257-280, 1994.

Holland, J. H., *Adaptation in Natural Artificial System*, University of Michigan Press, Ann Arbor, MI, 1975.

Huang, Y.-W et al., Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 16 (4), 2006.

IME, UERJ, UFRJ, UnB, Projeto H.264 – SETUP. *Linhas Mestras para Operação e Configuração de Sistemas Compressão de Vídeo para o SBTVD*, 2010.



ITU-R, Recommendation ITU-R BT.500-11: Methodology for the subjective assessment of the quality of television pictures. Swiss, 2002.

ITU-T. ITU-T Recommendation H.264. Advanced video coding for generic audiovisual services. 2007.

Jaumard , B., Hansen , P., Poggi di Aragao , M., Column generation methods for probabilistic logic. ORSA J. Comput.3, 135-148, 1991.

Kelly, J. P., Golden, B. L., Assad, A., A. Large-scale controlled rounding using tabu search with strategic oscillation. Ann. Ops Res. 41, 69-84, 1993.

Koumaras, H., Kourtis, A., Martakos, D., Evaluation of video quality based on objectively estimated metric. Journal of Communications and Networks, Vol.7, No.3, 2005.

Laguna, M. J., Barnes, W., Glover, F., Tabu search methods for a single machine scheduling problem. J. Intell. Manufac. 2, 63-74, 1991.

Laguna, M., Glover , F., Integrating target analysis and tabu search for improved scheduling systems. Expert SystemsApplic. 6, 287-298, 1993.

Laguna, M., Gonzalez-Verlarde, J. L., A search heuristic for just-in-time scheduling in parallel machines'. J. Intell. Manufac. 2, 253-260, 1991.

Malek, M., Guruswamy, M., Pandya , M., Owens , H., Serial and parallel simulated annealing and tabu search algorithmsfor the travelling salesman problem. Ann. Ops Res. 21, 59-84, 1989.

Malvar, H.S., Hallapuro, A., Karczewicz, M., Kerofsky, L., Low-complexity transform and quantization in H.264/AVC. IEEE transaction on Circuits and Systems for video technology, vol. 13, pp. 598–603, 2003.

Manoel, E. T. M., Codificação de Vídeo H.264 – Estudo de codificação mista de macroblocos. 2007. Dissertação (Mestrado em Engenharia Elétrica). Universidade Federal de santa Catarina, Florianópolis, 2007.

Michalewicz, Z., Genetic Algorithm + Data Structure = EvolutionPrograms. Springer, 1996.

Mooney, E. L., Rardin , R. L., Tabu search for a class of scheduling problems. Ann. Ops Res. 41, 253-278, 1993.

Moriyoshi, T., Shinohara, H., Miyakazi, T., Kuroda, I., Real-time software video codec with a fast adaptive motion vector search. Journal of VLSI Signal Processing, vol. 29, pp. 239–245, 2000.

Müller, G. I., Gómez, A. T. Utilização da Busca Tabu para a Geração de um Modelo Aplicado ao *Jobshop Scheduling Problem* Considerando um Sistema de Manufatura Flexível. Journal of Computer Science, vol. 6, p. 90-99, 2006.

Nemethova, O., Ries, M., Siffel, E., Rupp, M., Quality Assessment for H.264 Coded Low-Rate and low-Resolution Video Sequences. Proc. of Conf. on Internet and Inf.

- Technologies (CIIT), St. Thomas, US Virgin Islands, pp. 136-140, 2004.
- Oliveira, S., Stroud, G., A parallel version of tabu search and the path assignment problem. *Heuristics Combinat.Optimiz.* 4, 1-24, 1989.
- Osman, I. H., Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Ann. OpsRes.* 41, 421-452, 1993.
- Pinson, H. M., Wolf, S., A new standardized method for objectively measuring video quality. Institute for Telecommunication Science, Boulder CO 80305 USA, 2003.
- Richardson, I. E. G., H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia. UK: John Wiley & Sons, 2003.
- Richardson, I. Video Coding Design - Developing Image and Video Compression Systems. Chichester: John Wiley and Sons, 2002.
- Ries, M., Crespi, C., Nemethova, O., Rupp, M., Content Based Video Quality Estimation for H.264/AVC Video Streaming, Proc. of the IEEE WCNC'07, Hong Kong, 2007.
- Ries, M., Video Quality Estimation for Mobile Video Streaming. 2008. Dissertação (Mestrado em Engenharia). Tampere University of Technology, Finlândia, 2008.
- Semet, F., Taillard, E., Solving real-life vehicle routing problems efficiently using taboo search. *Ann. Ops Res.* 41, 469-488, 1993.
- Sikora, T., Trend and Perspectives in Image and Video Coding. Proceedings of IEEE, vol. 93, 2005.
- Skorin-Kapov, J., Tabu search applied to the quadratic assignment problem. *ORSA J. Comp.* 2, 33-45, 1990.
- Sullivan, G. J., Wiegand, T., Video Compression – From Concepts to the H.264/AVC Standard, Proceedings of the IEEE, vol. 93, no. 1, pp. 18-31, 2005.
- Sun, M., McKeown, P. G., Tabu search applied to the general fixed charge problem. *Ann. Ops Res.* 41, 405-420, 1993.
- Taillard, E., Some efficient heuristic methods for the flow shop sequencing problem. *Eur. J. Ops Res.* 47, 65-74, 1990.
- Taillard, E., Taboo search for the quadratic assignment problem. *Parallel Computing.* 17, 443-455, 1991.
- Talbi, E-G., Metaheuristics: from design to implementation. John Wiley & Sons, Inc., Hoboken, New Jersey, 1965, Ed. 2009.
- UNISINOS. Especificação de Software CODEC – Codificação e Decodificação de Sinais Fonte. Jul, 2008a.
- UNISINOS. Especificação Técnica de Hardware – Codificação e Decodificação de Sinais Fonte. 2008b.
- Wang, Z.; Sheikh, H. R.; Bovik, A. C., Objective Video Quality Assessment. The

Handbook of Video Databases: Design and Applications, B. Furht and O. Marqure, ed., CRC press, pp. 1041 – 1078, 2003.

Widmer , M., Hertz A., A new method for the flow sequencing problem. Eur. J. Ops Res. 41, 186-193, 1989.

Widmer, M., Job shop scheduling with tooling constraints: a tabu search approach. J. Opl Res. Soc. 24, 75-82, 1991.

Winkler, S.; Mohandas, P., *The evolution of video quality measurement: from PSNR to hybrid metrics*. IEEE Transactions on Broadcasting, pp. 660 – 668, 2008.

Wolf, S., Pinson, M. H., Application of the NTIA general video quality metric (VQM) to HDTV quality monitoring. Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM-07), Scottsdale, Arizona, 2007

Woodruff , D. L., Spearman , M. L., Sequencing and batching for two classes of jobs with deadlines and setup times. Prod. Ops Mangmnt 1, 87-102, 1992.

Yasakethu, S. L. P., Fernando, W. A. C., Adedoyin, S., Kondoz , A., A Rate Control Technique for Off Line H.264/AVC Video Coding Using Subjective Quality of Video. IEEE Transactions on Consumer Electronics, Vol. 54, No. 3, 2008

Yuelel, X., Duyan, B., Baixin, M., A Genetic Search Algorithm for Motion Estimation. IEEE, Signal Processing Proceedings, 2000.

Zhang, J., He, Y., Yang, S., Zhong, Y.; Performance and complexity joint optimization for H.264 video coding. ISCAS'03, vol. 2, pp. 888–891, 2003.

## Glossário

### Crominância ou Croma

Componente que adiciona cor a uma figura. No formato YUV, existem duas componentes de crominância, vermelha (Cr) e azul (Cb), onde Cb é a diferença entre o azul e a luminância (B-Y) e o Cr é a diferença entre o vermelho e a luminância (R-Y).

### Erro Residual

A diferença entre os valores preditos e os valores decodificados das amostras. Equivale também à diferença entre os valores preditos e os valores originais, encadeados aos seguintes processos: transformação (em coeficientes), quantização (truncamento), quantização inversa e transformação inversa.

### Fatia ou slice

Um conjunto de MB de um mesmo quadro, numerados seqüencialmente, que podem ser codificados isoladamente dos MBs das outras fatias do mesmo quadro. Cada fatia pode ser do tipo I, P, B, SI ou SP – restringindo os possíveis tipos de seus MBs.

### Fluxo de Bits (Bitstream)

Uma seqüência de bits que adota a sintaxe especificada pelo padrão H.264 e que codifica um determinado vídeo.

Formato YUV: formato de representação que divide a imagem em três componentes: um de luminância e dois de crominância.

### Frame (ou quadro)

Uma seqüência de vídeo é composta, na verdade, por uma coleção de “fotos” estáticas em uma taxa muito rápida, dando a sensação de movimento. Cada uma destas fotos é chamada de frame ou quadro.

### Frame B

É o frame que pode ser decodificado usando *inter prediction* ou *intra prediction* e as referências podem ser tanto de frames passados quanto futuros.

### Frame I

É o frame que foi codificado usando *intra prediction*, ou seja, não depende de outros frames para ser decodificado.

### Frame P

É o frame que pode ser decodificado usando *intra-prediction* ou *inter-prediction* usando no máximo um vetor de movimento e índice de referência para prever os valores de amostras de cada bloco.

### JIGA

É um sistema (hardware e software) para a realização de testes completos para o CODEC de vídeo, que envolve a geração de conteúdo, codificação/decodificação e visualização do vídeo decodificado.

### Luminância ou Luma

Componente de uma imagem que representa os níveis de luminosidade, ou seja, a intensidade de brilho dos diferentes pixels. Uma imagem em preto e branco na verdade é composta somente pela componente de luminância.

### Macrobloco (MB)

Unidade em que é dividido um frame do vídeo para facilitar o processamento. Um macroblock (MB) representa uma região quadrada de 16x16 pixels e ainda pode ser dividido. Pode-se ter sub-blocos de vários tamanhos: 16x8, 8x16, 8x8, 8x4, 4x8 e 4x4. Em alguns casos, é mais vantajoso processar blocos maiores ou macroblocks inteiros; porém, às vezes, é mais interessante o uso de blocos menores (8x4, por exemplo).

### Partição

Um bloco de amostras resultantes do particionamento de um submacrobloco para a predição inter.

### Pixel

A palavra pixel deriva de "picture element". Pixel é um ponto indivisível de uma imagem digital, o que equivale a uma amostra luma junto com as subamostras croma correspondentes.

### Referência Temporal

Parâmetro que seleciona qual dos quadros previamente decodificados é utilizado para a compensação de movimento de um bloco.

### Taxa de Bits

Quantidade de bits necessária para codificar um segundo de vídeo (geralmente em kbps – kilobits por segundo).

### Taxa-Distorção (TD)

Conjunto de medidas de desempenho de codificação que são inversamente proporcionais à taxa de bits e à distorção. Uma dessas medidas é o custo lagrangiano.