

**UNIVERSIDADE FEDERAL DE PELOTAS**  
**Centro de Desenvolvimento Tecnológico**  
Programa de Pós-Graduação em Computação



**Dissertação**

**RDE-AGMO: Seleção Automática de Pontos de Controle Eficientes para  
Codificadores HEVC Utilizando Algoritmos Genéticos**

**Italo Dombrowski Machado**

**Pelotas, 2020**

**Italo Dombrowski Machado**

**RDE-AGMO: Seleção Automática de Pontos de Controle Eficientes para  
Codificadores HEVC Utilizando Algoritmos Genéticos**

Dissertação apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Mestre em Ciência da Computação

Orientador: Prof. Dr. Bruno Zatt  
Coorientador: Prof. Dr. Marcelo Schiavon Porto  
Colaborador: Prof. Dr. Daniel Munari Vilchez Palomino

Pelotas, 2020

Universidade Federal de Pelotas / Sistema de Bibliotecas  
Catalogação na Publicação

M149r Machado, Italo Dombrowski

RDE-AGMO : seleção automática de pontos de controle eficientes para codificadores HEVC utilizando algoritmos genéticos / Italo Dombrowski Machado ; Bruno Zatt, orientador ; Marcelo Schiavon Porto, Daniel Munari Vilchez Palomino, coorientadores. — Pelotas, 2020.

97 f.

Dissertação (Mestrado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2020.

1. HEVC. 2. AGMO. 3. Problema de otimização. 4. Controle de energia. I. Zatt, Bruno, orient. II. Porto, Marcelo Schiavon, coorient. III. Palomino, Daniel Munari Vilchez, coorient. IV. Título.

CDD : 005

Eu dedico este trabalho a toda minha família, amigos e professores que me inspiraram e apoiaram ao longo da minha vida acadêmica.

## **AGRADECIMENTOS**

Primeiramente agradeço a toda minha família que me apoiou e deu suporte emocional e financeiro para iniciar o meu mestrado e concluir com esta dissertação. Gostaria de agradecer também em particular meu primo Agemir Marcolin por como um irmão ter estado ao meu lado proferindo palavras e conselhos sábios ao longo desta jornada.

À todos os meus professores envolvidos na minha vida acadêmica que me proporcionaram o caminho para atingir o sucesso neste trabalho de mestrado. Em especial o professor Marilton Sanchotene cuja aula me inspirou a ideia de desenvolver o algoritmo apresentado neste trabalho. E também ao professor Bruno Zatt, que o auxílio tornou possível a conclusão deste trabalho e a inteligência fez-me sentir sobre os ombros de gigantes.

À Universidade Federal de Pelotas, que apesar de todos os cortes de verbas e incertezas políticas segue fazendo o máximo possível pelo benefício de seus estudantes. E a CAPES por financiar minha pesquisa através de uma bolsa de mestrado.

Por fim, agradeço também a todos aqueles que estiveram presentes na minha vida me proporcionando experiências positivas porém que infelizmente não puderam em particular nestes agradecimentos.

*A menos que modifiquemos a nossa maneira de pensar, não seremos capazes de resolver os problemas causados pela forma como nos acostumamos a ver o mundo — ALBERT EINSTEIN*

## RESUMO

DOMBROWSKI MACHADO, Italo. **RDE-AGMO: Seleção Automática de Pontos de Controle Eficientes para Codificadores HEVC Utilizando Algoritmos Genéticos.** 2020. 97 f. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2020.

Devido ao alto consumo energético do processo de codificação de vídeo dos codificadores HEVC e ao fato de que estes deverão ser executados em bilhões de dispositivos móveis que possuem reservas de energia limitadas, é de extremo interesse reduzir o consumo energético destes softwares. Para isto, foram propostos controladores de energia que gerenciam o consumo energético de codificadores HEVC através da seleção dinâmica de diferentes Pontos de Controle (PCs). Apesar da tarefa de avaliar e selecionar este conjunto de PCs ser uma complexa otimização multiobjetivo e crucial para a eficiência do controle, as metodologias aplicadas para realizar esta análise utilizam técnicas de seleção que possuem aspectos que podem ser melhorados. Pois as atuais metodologias de análise de sensibilidade necessitam que um especialista crie e selecione manualmente os PCs, o que impossibilita a avaliação de um conjunto aceitável de configurações e que o desenvolvedor escolha com precisão a heterogeneidade no consumo de seus PCs. Desta forma, esta dissertação apresenta o RDE-AGMO do inglês *Rate-Distortion-Energy*, um algoritmo genético multiobjetivo (AGMO) capaz de selecionar configurações eficientes e com consumos energéticos específicos para codificadores de vídeo. O RDE-AGMO é capaz de realizar o processo de análise de sensibilidade de forma mais veloz, autônoma e abrangente do que as metodologias atualmente utilizadas enquanto proporciona benefícios ao comportamento do controlador. Além disto, devido a generalidade do algoritmo, este pode ser aplicado a qualquer codificador e variável de controle (energia, tempo, complexidade aritmética) sem precisar de grandes modificações. Ao observarmos o impacto da utilização do RDE-AGMO no desenvolvimento de um controlador foi possível notar que houve uma melhora na distribuição energética da sua tabela de PCs, com o erro médio quadrático das configurações diminuindo de 41% para 16%. Esta melhor distribuição no consumo dos PCs conseqüentemente aumentou a faixa de controle possível de um máximo de 30% de redução energética para 70%. Ademais, o BD-BR médio causado pelo controle energético caiu de 4,07% para 3,45% em troca de um acréscimo no erro do consumo obtido em relação ao desejado de apenas 2,45%.

**Palavras-Chave:** hevc; agmo; problema de otimização; controle de energia

## ABSTRACT

DOMBROWSKI MACHADO, Italo. **RDE-AGMO: Automatic Selections of Efficient Control Points for Video encoders Utilizing Multiobjective Genetic Algorithms.** 2020. 97 f. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2020.

Due to the high energy consumption of the video encoding process of HEVC encoders and the fact that they must be executed on billions of mobile devices that have limited energy reserves, it is of extreme interest to reduce the energy consumption of these applications. Aiming to achieve reduction, energy controllers that manage the energy consumption of HEVC encoders have been proposed. Thereunto, these controls have a set of configurations called Control Points (PCs) that are dynamically selected to reduce or increase their consumption when necessary. This set of PCs is extremely important for the performance of the controller and is assembled through the sensitivity analysis process. Despite this being a complex task of multiobjective optimization and crucial for the efficiency of the controller, the methods applied to carry out this technical selection analysis presents aspects that can be improved. Since the current sensitivity analysis methodologies require an expert to manually create and select PCs, it is impossible to evaluate an acceptable set of configurations and for the developer to accurately choose the consumption heterogeneity of their PCs. Therefore, this dissertation presents the RDE-AGMO *Rate-Distortion-Energy* Multiobjective Genetic Algorithm, a multiobjective genetic algorithm (MOGA) capable of selecting efficient configurations with specific energy consumptions for video encoders. The RDE-AGMO is able to carry out the sensitivity analysis process in a faster, autonomous and comprehensive way than the currently used methodologies while providing benefits to the controller's behavior. Furthermore, due to the generality of the algorithm, it can be applied to any coder and controller variable (energy, time, arithmetic complexity) without requiring major modifications. When observing the impact of the RDE-AGMO applied in the development of a controller, it was possible to notice that there was an improvement in the energy distribution of its PC table, with the mean square error of the configurations decreasing from 41% to 16%. This better distribution in the consumption of PCs consequently increased the possible control range from a maximum of 30% of energy reduction to 70%. Additionally, the average BD-BR caused by the energy controller decreased from 4.07% to 3.45% in exchange for an increase in the consumption error obtained in relation to the desired of only 2.45%.

**Keywords:** hevc; moga; optimization problem; energy control

## LISTA DE FIGURAS

Figura 1	Diagrama de Blocos do Processo de Codificação . . . . .	20
Figura 2	Diagrama de Blocos do Sistema de Controle de Malha Fechada definido por (OGATA; YANG, 2002) . . . . .	24
Figura 3	Diagrama de Blocos do Sistema de Controle em codificadores HEVC	25
Figura 4	Seleção Via Frente de Pareto . . . . .	32
Figura 5	Fluxograma de um Algoritmo Genético . . . . .	35
Figura 6	Distribuição estatística da seleção de indivíduos no método de roleta	36
Figura 7	<i>Crossover</i> de Ponto Único . . . . .	37
Figura 8	<i>Crossover</i> de Dois Pontos . . . . .	37
Figura 9	<i>Crossover</i> Uniforme . . . . .	37
Figura 10	Funcionalidade e Aplicação do RDE-AGMO . . . . .	40
Figura 11	Representação de um indivíduo . . . . .	41
Figura 12	Distribuição RDE da Primeira População . . . . .	46
Figura 13	Passos do Operador de Seleção . . . . .	48
Figura 14	Operador de <i>Crossover</i> Uniforme . . . . .	49
Figura 15	Desvio Padrão das Gerações Buscando o PC30 para o Vídeo PartyScene . . . . .	54
Figura 16	Média de Consumo das Gerações Buscando o PC50 para o Vídeo PartyScene . . . . .	55
Figura 17	Média de Consumo das Gerações Buscando o PC90 para o Vídeo PartyScene . . . . .	55
Figura 18	Desvio Padrão das Gerações Buscando o PC20 para o Vídeo Kimono	56
Figura 19	Consumo Médio das Gerações Buscando o PC20 para o Vídeo Kimono . . . . .	57
Figura 20	Consumo Médio das Gerações Buscando o PC60 para o Vídeo Kimono . . . . .	57
Figura 21	Comportamento do Controle de MACHADO (2017) com os PCs Encontrados pelo RDE-AGMO (BasketballDrill QP 32 SP 30) . . . . .	69
Figura 22	Comportamento do Controle de MACHADO (2017) com os PCs Originais (BasketballDrill QP 32 SP 30) . . . . .	69
Figura 23	Comportamento do Controle com os PCs Encontrados pelo RDE-AGMO sob Mudança de SP30 para SP60 (BasketballDrill QP 32) . . . . .	70
Figura 24	Comportamento do Controle com os PCs Originais Sob Mudança de SP30 para SP60 (BasketballDrill QP 32) . . . . .	70

## LISTA DE TABELAS

Tabela 1	Parâmetros de Relacionados as CTUs, CUs e Modo de Decisão . . .	21
Tabela 2	Parâmetros de Relacionados ao Particionamento das PUs e TUs . .	22
Tabela 3	Parâmetros Relacionados as Predições Inter-quadros e Intra-quadro	23
Tabela 4	Parâmetros Relacionados aos Tipos de Quadros . . . . .	23
Tabela 5	Tabela de Parâmetros Avaliados por GRELLERT (2014) . . . . .	26
Tabela 6	Tabela de PCs do Controlador de MACHADO (2017) . . . . .	29
Tabela 7	Parâmetros avaliados com suas respectivas configurações . . . . .	50
Tabela 8	Diferentes avaliações para cada aspecto do algoritmo . . . . .	51
Tabela 9	Parâmetros dos PCs Encontrados para vídeos 448p . . . . .	58
Tabela 10	Comparação Entre a Distribuição da Economia dos PCs Propostos pelo RDE-AGMO e dos Demais Controladores . . . . .	60
Tabela 11	BD-BRs e Consumos dos Pontos de Controle Encontrados para CTCs 448p . . . . .	61
Tabela 12	Parâmetros dos PCs selecionados pelo RDE-AGMO para o Vídeo Kimono . . . . .	62
Tabela 13	Comparação Entre a Distribuição da Economia dos PCs Propostos pelo RDE-AGMO e dos Controladores . . . . .	63
Tabela 14	BD-BRs das Configurações Encontradas para o Vídeo Kimono . . .	64
Tabela 15	RMSEs Causados por Ambos CSs para Todos Casos de Teste . . .	65
Tabela 16	Redução Energética Atingida por Ambos CSs para Todos Casos de Teste . . . . .	66
Tabela 17	Comparação de BD-BRs dos Casos de Teste para Ambos CSs . . .	67

## LISTA DE ABREVIATURAS E SIGLAS

AE	Algoritmos Evolucionários
AMP	Asymmetric Motion Partition
BD	Bjøntegaard
BD-BR	Bjøntegaard Bit-rate
BD-PSNR	Bjøntegaard Peak Signal-to-Noise Ratio
CH	Crossover Heurístico
CI	Crossover Intermediário
CS	Configuration Set
CTU	Coding Tree Unit
CA	Complexidade Aritmética
CFG-LUT	Configuration Lookup Table
CRFC	Computação do Fator de Redução de Complexidade
CTU-HT	CTU History Table
CU	Coding Unit
GOP	Group of Pictures
HME	Hadamard Motion Estimation
HEVC	High Efficiency Video Coding
HM	HEVC Test Model
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector
JCT-VC	Joint Collaborative Team on Video Coding
ME	Motion Estimation
MGTC	Medium Granularity Time Control
MSR	Model-Specific Registers
MSE	Mean Squared Error
PC	Ponto de Controle

PID	Proportional Integral Derivative
PO	Problema de Otimização
POMO	Problema de Otimização Multiobjetivo
PCM	Phase-Change Memory
PS	Parameter Set
PU	Prediction Unit
QP	Quantization Parameter
RAPL	Running Average power Limit
RDO	Rate Distortion Optimization
RDE	Rate-Distortion-Energy
SAO	Sample Adaptive Offset
SCMF	Sistema de Controle de Malha Fechada
SGA	Simple Genetic Algorithm
SO	Sistema Operacional
SP	SetPoint
SR	Search Range
TPC	Tabela de Pontos de Controle
FGTC	Fine Granularity Time Control
TU	Transform Unit
$WN_{ERD}$	Weighted Normalized Energy Compression Distortion

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
1.1	Problema de Pesquisa	15
1.2	Objetivo	17
1.3	Estrutura do Texto	18
<b>2</b>	<b>PADRÃO HEVC E TRABALHOS RELACIONADOS</b>	<b>19</b>
2.1	Estrutura de Codificação	19
2.2	Codificador HEVC x265	21
2.3	Controle de Energia em Codificadores HEVC	23
2.3.1	Trabalhos Relacionados	25
2.3.2	Desafio de Pesquisa	29
<b>3</b>	<b>PROBLEMAS DE OTIMIZAÇÃO MULTIOBJETIVO E ALGORITMOS GENÉTICOS</b>	<b>31</b>
3.1	Seleção	34
3.2	Crossover	37
3.3	Mutação	39
<b>4</b>	<b>RDE-AGMO: SELETOR DE CONFIGURAÇÕES EFICIENTES PARA CODIFICADORES</b>	<b>40</b>
4.1	Modelagem e Definição dos Operadores Genéticos	41
4.2	Seleção	47
4.3	Crossover e Mutação	49
4.4	Mutação	50
<b>5</b>	<b>RESULTADOS</b>	<b>52</b>
5.1	Comportamento e Convergência do RDE-AGMO	53
5.2	Eficiências dos Pontos de Controle	57
5.3	Impacto das Configurações no Controlador de Energia	64
<b>6</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS</b>	<b>71</b>
6.1	Trabalhos Futuros	73
	<b>REFERÊNCIAS</b>	<b>74</b>
<b>APÊNDICE A</b>	<b>COMPORTAMENTO DO CONTROLE COM AS CONFIGURAÇÕES ENCONTRADAS VIDEOS CLASSE C SP30</b>	<b>78</b>
<b>APÊNDICE B</b>	<b>COMPORTAMENTO DO CONTROLE COM AS CONFIGURAÇÕES ENCONTRADAS VIDEOS CLASSE C SP70</b>	<b>83</b>

<b>APÊNDICE C</b>	<b>COMPORTAMENTO DO CONTROLE COM AS CONFIGURAÇÕES ENCONTRADAS VIDEOS CLASSE B SP30 . . . . .</b>	<b>88</b>
<b>APÊNDICE D</b>	<b>COMPORTAMENTO DO CONTROLE COM AS CONFIGURAÇÕES ENCONTRADAS VIDEOS CLASSE B SP70 . . . . .</b>	<b>93</b>

# 1 INTRODUÇÃO

O veloz desenvolvimento tecnológico que vem ocorrendo nas últimas décadas proporcionou grande capacidade computacional e inúmeras funcionalidades para plataformas como computadores pessoais, *vídeo games* e celulares. Tal desenvolvimento transformou estes dispositivos em ferramentas indispensáveis no dia a dia das pessoas, acarretando em uma alta popularização dos mesmos. De acordo com uma projeção realizada por CISCO (2020), o número de aparelhos móveis superará a população mundial em cinco anos. A união dos aumentos da capacidade computacional e da utilização destes dispositivos possui um impacto substancial no consumo de energia elétrica no mundo.

Além disso, plataformas como celulares, câmeras digitais e *laptops*, entre outros, possuem uma fonte de energia limitada, pois funcionam através de baterias para manterem a sua portabilidade, que é uma de suas características essenciais. Frente a ambos contextos, é de extrema relevância reduzir, dentro do possível, o consumo energético de tais dispositivos, não só para aumentar sua autonomia, mas também para contribuir na redução do consumo energético mundial.

Neste contexto, a popularização e utilização de vídeos digitais em dispositivos móveis se tornou evidente. Os avanços tecnológicos e a popularização de vídeos digitais viabilizaram que dispositivos móveis possam gravar vídeos com resoluções muito altas como 4k (3840 x 2160 pixels). Para acompanhar este crescimento nas resoluções dos vídeos, foi necessário o desenvolvimento de um novo padrão de codificação de vídeo que atingisse maiores taxas de compressão sem causar perdas de qualidade significativas. Este novo padrão denominado *High Efficiency Video Coding* (HEVC) foi desenvolvido pela *Joint Collaborative Team on Video Coding* (JCT-VC) e é atualmente considerado o estado da arte em codificação de vídeo (ITU-T, 2015).

Esta busca por maiores taxas de compressão acarretou também em um acréscimo significativo no custo computacional do processo de codificação deste novo padrão. Este custo computacional implica consequentemente em um grande custo energético, o que é indesejável em tais aplicações, especialmente por estas ocorrerem com alta frequência em dispositivos cuja disponibilidade energética é restrita por ser proveni-

ente de baterias.

Devido a energia disponível em dispositivos móveis ser limitada, é interessante podermos alterar o consumo do processo de codificação dinamicamente de acordo com a necessidade. Para proporcionar esta importante dinamicidade aos codificadores HEVC, diversos trabalhos propuseram metodologias para a realização de controle de complexidade e energia em diferentes codificadores deste padrão. Contudo, o desenvolvimento de um controlador de energia não é uma tarefa trivial, em vista que múltiplas etapas do processo demandam análises complexas e extensivas sobre o codificador e o funcionamento do controle perante o mesmo.

## 1.1 Problema de Pesquisa

Segundo uma pesquisa realizada por CISCO (2020), vídeos serão responsáveis por 79% do tráfego de dados em dispositivos móveis até 2022. Com este dado é possível afirmar que grande parte dos vídeos digitais são utilizados em aparelhos que dependem de bateria para o seu funcionamento. Desta forma, torna-se crucial a realização de controle de energia em codificadores de vídeo.

Na literatura existem múltiplos trabalhos que realizam controle em codificadores HEVC. Entre estes trabalhos estão CORRÊA (2016), PENNY (2017) e MACHADO (2017) que utilizam metodologias similares para o desenvolvimento e atuação do controle. Com o último sendo um trabalho desenvolvido por mim previamente a este. Contudo, embora a análise de sensibilidade seja uma das etapas mais importantes no desenvolvimento do controlador, as metodologias empregadas utilizam técnicas que deixam espaço para melhorias.

O processo de análise de sensibilidade é responsável por definir os pontos de operação que serão utilizados pelo controlador, possuindo assim grande influência na eficiência e no comportamento do mesmo. Para definir estes pontos é necessário codificar vídeos com diversas configurações do codificador para assim identificar o impacto de cada ferramenta/configuração no consumo. Considerando-se que um codificador pode ter mais de 30 parâmetros que combinados resultam em aproximadamente 280 milhões de configurações possíveis (através da permutação entre todos os parâmetros de codificação), e que levaria cerca de 13 anos para testar todas as configurações assumindo que em média cada codificação levaria 5 minutos, podemos afirmar que é praticamente inviável avaliar todas as configurações possíveis. Além disto, as configurações possuem impacto em dois eixos interdependentes, pois uma economia no consumo energético sempre possui certo impacto na eficiência de codificação (eficiência de codificação que já reflete outros dois eixos, qualidade e taxa de bits) (PENNY, 2017). Tal dependência entre estas duas variáveis torna a busca por configurações energeticamente eficientes em codificadores uma tarefa desafiadora.

Apesar destas características definirem a análise de sensibilidade como um problema de otimização multiobjetivo (POMO) (DEB, 2014) com um conjunto incontável de soluções por ter de avaliar dois eixos interdependentes (eficiência e energia), os trabalhos presentes na literatura utilizam metodologias que necessitam de um especialista na área para selecionar empiricamente um conjunto de configurações para serem avaliadas ao invés de utilizar uma metodologia de busca apropriada para POMOs que possibilite a avaliação de um grande número de soluções. A qualidade desta busca empírica por configurações eficientes depende da competência e experiência do especialista, o que não é ideal em vista das limitações cognitivas humanas e que em muitos cenários não é possível ter-se um bom especialista à disposição. Adicionalmente, os impactos de diferentes parâmetros de codificação não apenas variam consideravelmente, como também podem possuir grandes influências nos comportamentos uns dos outros, fazendo com que, ao se utilizar as técnicas atuais, seja necessário repetir a busca caso se falhe ao prever a interação dos comportamentos de diferentes parâmetros. Além disto, mesmo que bem sucedida, a busca serviria apenas para o cenário em estudo e não poderia ser aplicada em outras circunstâncias como em outros codificadores. Estas características reforçam não somente a alta complexidade da busca por configurações eficientes, mas também a necessidade de uma metodologia de busca bem definida. Após efetuar a seleção, as configurações mais eficientes são identificadas através da técnica de Ótimo de Pareto. Tendo em vista que esta técnica determinística apenas seleciona os pontos ótimos dentro de um conjunto de soluções, é possível afirmar que sua eficiência é extremamente dependente da qualidade da busca por configurações realizada previamente. Por fim, a utilização desta metodologia não garante uma heterogeneidade na economia energética das configurações selecionadas, permitindo assim que múltiplos pontos de operação tenham consumos muito semelhantes. Esta homogeneidade entre o consumo dos pontos de operação do controlador impossibilitam que este atue de forma adequada em cenários que careçam de diferentes níveis de economia, como por exemplo, em vídeos com características muito dinâmicas.

Sendo assim, o principal problema de pesquisa para o desenvolvimento deste trabalho consiste na busca por uma metodologia de seleção de configurações energeticamente eficientes para codificadores HEVC. Desta forma, torna-se necessário um algoritmo que não somente automatize o processo de análise de sensibilidade mas também viabilize a exploração de um número muito maior de configurações, excluindo a necessidade de especialistas e permitindo uma seleção heterogênea das configurações. Por fim, o algoritmo desenvolvido ainda seria genérico, podendo assim ser aplicado a outras aplicações, a qualquer codificador HEVC e ao processo de decodificação.

Devido a alta dificuldade de se encontrar soluções ótimas para POMOs, existem di-

versos algoritmos e técnicas que tentam solucionar tais problemas e podem ser aplicados ao contexto de análise de sensibilidade de codificadores HEVC. Ainda, de acordo com SOARES (1997) uma das técnicas de busca para POMOs que possuem maior eficiência são os algoritmos genéticos. Os AGs vem sendo utilizados com sucesso e diferentes áreas como otimização multiobjetivo de fontes de energia em prédios (ROBYR et al., 2018), de memórias PCM (*Phase-Change Memory*) para *green computing* na nuvem (QIU et al., 2015), decisões de bancos para empréstimos (METAWA; HASSAN; ELHOSENY, 2017) e diagnósticos de falhas em engrenagens (CERRADA et al., 2016).

## 1.2 Objetivo

Este trabalho tem como objetivo principal desenvolver um Algoritmo Genético (AG) que substitua e automatize a etapa de análise de sensibilidade presente no desenvolvimento de controladores de energia para codificadores HEVC. A técnica proposta deverá analisar milhares de configurações sem a dependência de um especialista, permitir a seleção de configurações de consumo heterogêneas entre si e diminuir o tempo do processo da análise de sensibilidade de uma escala de meses para semanas devido à sua automaticidade. Além disto, o AG proposto poderá ser facilmente aplicado a qualquer codificador HEVC, cenários e variáveis dentro do contexto de codificação de vídeo, como por exemplo o processo de decodificação ou variáveis como energia, tempo e complexidade aritmética.

Como este trabalho tem um viés aplicado a sistemas móveis, o AG será implementado em um codificador de vídeo comercial otimizado. Por ser um codificador HEVC amplamente utilizado, será aplicado o codificador x265 (MULTICOREWARE, 2016) que possui um bom compromisso entre eficiência de codificação e velocidade de codificação. Além disto foi utilizado no trabalho MACHADO (2017) previamente desenvolvido à este e mostrou-se um codificador mais complexo de se encontrar configurações eficientes quando comparado com o HM utilizado em PENNY (2017). Tais características o favorece como base para análise de eficiência do algoritmo e para realizar comparações com outros trabalhos.

Objetivos específicos:

- Encontrar configurações eficientes para diferentes níveis de economia;
- Encontrar PCs com economias próximas do desejado;
- Diminuir o impacto do controle desenvolvido por MACHADO (2017);
- Otimizar o comportamento das atuações do controle.

### 1.3 Estrutura do Texto

Este trabalho está dividido em seis capítulos. O segundo capítulo está dividido em três seções, na primeira é realizada uma revisão sobre a estrutura de codificação do padrão HEVC. Na segunda seção as ferramentas e parâmetros de codificação do codificador x265 são explicadas. Por fim, na terceira seção o capítulo aborda o funcionamento dos controladores desenvolvidos para codificadores HEVC, assim como um estudo sobre as metodologias utilizadas nos trabalhos que desenvolvem controladores para codificadores HEVC assim como o desafio de pesquisa presente no desenvolvimento destes trabalhos.

No terceiro capítulo será realizada uma revisão sobre os conceitos envolvendo problemas de otimização multiobjetivos e algoritmos genéticos. Em seguida será dado maior detalhamento sobre as etapas e características dos algoritmos genéticos.

No quarto capítulo o estudo das diferentes técnicas avaliadas para cada operador genético do AGMO desenvolvido e o funcionamento do mesmo serão explicados detalhadamente.

No quinto capítulo primeiramente serão apresentados os resultados do comportamento do algoritmo desenvolvido em busca de diferentes pontos de controle com consumos específicos. Por conseguinte, serão avaliadas as eficiências das configurações encontradas para então ser discutido o impacto da aplicação destas em um controlador de energia.

Por fim, no capítulo seis apresenta as conclusões deste trabalho e propostas de trabalhos futuros.

## 2 PADRÃO HEVC E TRABALHOS RELACIONADOS

Este capítulo apresenta na seção 2.1 um breve embasamento teórico e conceitos fundamentais do processo de codificação e da estrutura de funcionamento do padrão HEVC. O conteúdo discutido nesta seção é importante para uma boa compreensão sobre a codificação no padrão HEVC e sobre o trabalho proposto. No entanto, não é o objetivo deste trabalho abordar tais conceitos de forma profunda, mas para cada tema discutido serão referenciados trabalhos da literatura onde os conteúdos estão detalhados em um maior aprofundamento. Em seguida, na seção 2.2 serão abordados os parâmetros de codificação do codificador x265 que será utilizado no desenvolvimento deste trabalho. Por fim, na seção 2.3 será discutida a estrutura utilizada para a realização de controle de diferentes variáveis em codificadores HEVC, os trabalhos relacionados que aplicam esta estrutura e o desafio de pesquisa presente no desenvolvimento destes controles.

### 2.1 Estrutura de Codificação

Os codificadores do padrão HEVC respeitam um fluxo de operações para realizar a codificação (SZE, 2014). A Figura 1 representa o diagrama de blocos deste curso de operações omitindo os sinais de controle do codificador.

O primeiro passo a se codificar um vídeo é dividir seus quadros em GOPs (*Group of Pictures*). Um GOP define como os quadros Intra (I), Preditivos (P) e Bipreditivos (B) serão organizados, os valores de QP (*Quantization Parameters*) e quadros de referências utilizados para serem codificados (ROSEWARNE et al., 2015). Enquanto quadros I podem ser codificados apenas com a predição intra, os demais tipos podem utilizar tanto a predição intra-quadro como a inter-quadros. Intermitentemente, um quadro I é colocado como RID (Refrescamento Instantâneo do Decoder) para dizer que este não pode utilizar referências anteriores na codificação de quadros futuros. Este quadro RID é essencial para se navegar livremente ao longo de vídeos reproduzidos na internet por permitir que o vídeo possa ser carregado a partir deste quadro sem a necessidade buscar quadros de referência anteriores.

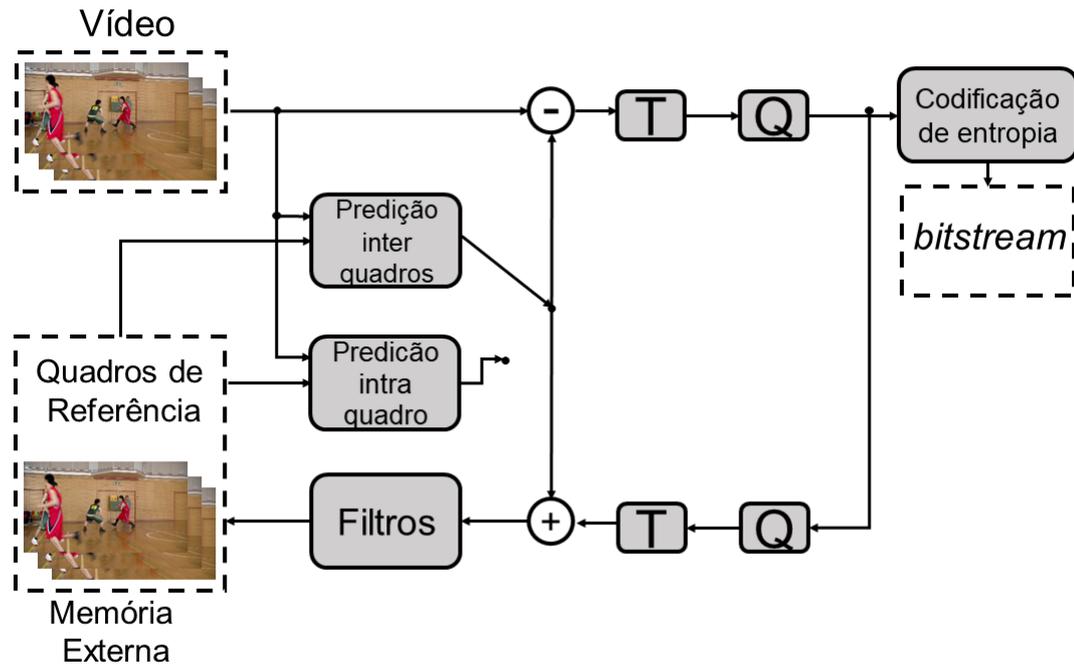


Figura 1 – Diagrama de Blocos do Processo de Codificação

Após a definição dos GOPs o vídeo é dividido em blocos denominados de CTUs (*Coding Tree Units*) que são submetidos ao laço de codificação representado pela Figura 1. Primeiramente a CTU a ser codificada é selecionada e particionada em blocos menores de acordo com um modo de decisão (SZE, 2014). Em seguida, cada um destes blocos é enviado para a etapa de predição que visa tirar proveito de informações redundantes nos quadros. Enquanto a predição intra-quadro explora redundâncias nos pixels do mesmo quadro, a inter-quadros explora redundâncias entre dois quadros temporalmente próximos. O quadro predito é então subtraído da CTU atual, resultando em um bloco de resíduos que são como um erro entre as informações preditas e as originais (RICHARDSON, 2002). Posteriormente, os resíduos passam pelas etapas de Transformada e Quantização. Na transformada as informações são levadas para o domínio das frequências onde então as frequências mais altas que são menos perceptíveis ao sistema visual humano são descartadas pela quantização (GONZALEZ R., 2003). As informações são então enviadas para o codificador de entropia, que por sua vez explora redundâncias no sinal e gera o *bitstream* final para ser armazenado ou transmitido (SZE, 2014). Em seguida, as informações geradas pela quantização também são enviadas para o estágio de decodificação, onde os resíduos são reconstruídos e adicionados às amostras preditas. Por conseguinte, o bloco reconstruído é filtrado para se remover artefatos gerados pela codificação. Por fim, esta informação é armazenada como quadro reconstruído para ser utilizado como referências nas próximas iterações do laço de codificação (RICHARDSON, 2002).

## 2.2 Codificador HEVC x265

Existem múltiplos codificadores que seguem as definições do padrão HEVC. Entre estes estão o HM (BOYCE, 2014) que é o modelo de referência do padrão, o KVZAAR (GROUP, 2020) um codificador acadêmico *open source* desenvolvido na universidade de Tampere e o x265 (MULTICOREWARE, 2016) que por sua vez é um codificador comercial desenvolvido pela empresa *MulticoreWare*. Neste trabalho será utilizado o codificador x265 por este já ser amplamente difundido no mercado e devido a grande eficiência e otimização das ferramentas implementadas neste codificador que nos dão um cenário mais próximo da realidade de uso do que codificadores acadêmicos. As estruturas de dados e ferramentas do padrão HEVC implementadas no x265 serão brevemente descritas a seguir e apresentadas em tabelas com suas respectivas configurações e influências no processo de codificação.

Para realizar este fluxo de codificação com maior performance e adaptação a múltiplas resoluções na sua codificação, o padrão HEVC definiu uma série de ferramentas, estruturas e particionamentos que diferem dos padrões anteriores. A maioria destas ferramentas e estruturas podem ser ativadas, desativadas ou configuradas através de parâmetros de codificação. O funcionamento destas ferramentas do codificador serão detalhados, e os parâmetros utilizados para configurá-los apresentados de acordo com o codificador x265.

Dentre estas estruturas estão as já mencionadas CTUs, que são semelhantes aos Macroblocos utilizados no padrão H.264/AVC (BROSS et al., 2011). Todo o quadro a ser codificado é dividido em múltiplas CTUs cujo tamanho máximo atualmente admitido é de 64x64 amostras. Para prover uma codificação mais flexível, cada CTU pode ser subdividida em quatro CUs (*Coding Units*) menores de tamanho 32x32. Ainda, o processo de divisão pode ser recursivamente repetido em cada CU até o tamanho mínimo de 8x8 ser atingido, formando uma árvore quaternária (SULLIVAN et al., 2012). Vale também ressaltar que as CUs não são sempre exploradas até seu menor tamanho devido às heurísticas implementadas no modo de decisão do padrão HEVC. Esta estratégia de divisões é interessante pois permite dividir o quadro em regiões com redundâncias similares que podem ser aproveitadas mais eficientemente na etapa de predição. Na Tabela 1 podemos observar os parâmetros que podem ser utilizados para alterar as partições das CTUs, CUs e as heurísticas do modo de decisão.

Tabela 1 – Parâmetros de Relacionados as CTUs, CUs e Modo de Decisão

Parâmetro	Função	Valores Assumidos
CTU	Limita o tamanho máximo das CTUs (altura e largura).	64   32   16
Cutrees	Ativa o uso de predição de vetores de movimento de baixa resolução para calibrar fatores de quantização adaptativa.	On/Off
Limit-modes	Limita os modos de partição testados para cada CU através de métricas de custo.	On/Off
Min-Cu-Size	Tamanho mínimo das CUs.	32   16   8
Rd-level	Limita o nível de RDO utilizado nos modos de decisão.	1...6
Rdoq-Level	Limita o uso da análise de Rate-Distortion na quantização.	0   1   2
Rskip	Permite/Proíbe a terminação prematura do recursão de profundidade das CUs.	On/Off

O padrão ainda define que durante o processo de predição cada CU pode ser dividida em PUs (*Prediction Units*), que por sua vez armazena diversas informações sobre o processo de predição, como vetores de movimento, índices dos quadros de referência, modo de predição, etc (SZE, 2014). Cada PU pode ser particionada de oito maneiras distintas, onde quatro destas maneiras são simétricas e as outras quatro assimétricas. Além das PUs o padrão ainda define que uma CU pode possuir uma ou mais TUs (*Transform Units*) que podem estar organizadas no formato de uma *quadtree* (ROSEWARNE et al., 2015). Esta estrutura pode assumir tamanhos de 32x32, 16x16, 8x8, 4x4 e é crucial nas etapas das transformadas e da quantização. A Tabela 2 apresenta os parâmetros de codificação relacionados as estruturas descritas neste parágrafo.

Tabela 2 – Parâmetros de Relacionados ao Particionamento das PUs e TUs

Parâmetro	Função	Valores Assumidos
AMP	Habilita/Desabilita o uso de PUs com partições assimétricas.	On/Off
Limit-TU	Heurística de particionamento para TUs codificadas pela predição inter-quadros.	0...4
Rect	Habilita/Desabilita o uso de PUs com partições retangulares.	On/Off
SignHide	Esconde um bit de sinal das amostras por TU.	On/Off
TU-intra-depth	Limita o particionamento das TUs codificadas pela predição intra-quadro.	1...4
TU-inter-depth	Limita o particionamento das TUs codificadas pela predição inter-quadros.	1...4

Além destas complexas estruturas, o padrão ainda conta com as predições inter-quadros e intra-quadro para diminuir a quantidade de informações redundantes presentes nos quadros SZE (2014). A predição inter-quadros busca encontrar redundâncias utilizando um ou mais quadros de referência temporalmente vizinhos. A distância temporal entre os vizinhos e o número de quadros de referência podem ser alterados para diminuir o custo da codificação em troca de eficiência. Além disto, por buscar semelhanças entre múltiplos quadros, esta ferramenta é responsável pelo maior custo computacional assim como pelos maiores ganhos de compressão da codificação (CORRÊA et al., 2012). Devido ao alto custo desta ferramenta, para tornar sua utilização mais viável as redundâncias são buscadas apenas em uma determinada área de busca em volta do bloco sendo predito. O tamanho desta área, assim como a forma com que as informações serão procuradas nela, podem ser alteradas através de parâmetros. Ademais, o padrão ainda possui a ferramenta de predição intra-quadro, que por sua vez utiliza blocos vizinhos já preditos para reduzir redundâncias presentes no mesmo quadro (GHANBARI, 2003). Os parâmetros utilizados para alterarem o funcionamento destas predições podem ser observados na Tabela 3.

Como foi previamente discutido, existem três classes que um quadro pode ser classificado. O primeiro tipo de quadro é o quadro-I, cujo possui o menor custo e eficiência de codificação por utilizar apenas a predição intra-quadros no seu processamento (SZE, 2014). Contudo, por utilizar apenas a predição intra, quadros de referência não são utilizados na sua codificação, o que é essencial em casos onde não há quadros

Tabela 3 – Parâmetros Relacionados as Predições Inter-quadros e Intra-quadro

Parâmetro	Função	Valores Assumidos
Early-skip	Termina a busca por blocos candidatos caso o bloco analisado não produza resíduos.	On/Off
Fast-intra	Aplica uma heurística para realizar a predição intra-quadro de forma mais rápida em troca de eficiência.	On/Off
Limit-refs	Limita o número de referências de acordo com a profundidade das CUs utilizadas.	0   1   2   3
MaxMerge	Limita o número máximo de vizinhos de blocos candidatos cujos vetores de movimento podem ser fundidos.	1...5
ME	Define o algoritmo de busca das amostras dentro da área de busca na predição inter-quadros.	1...5
MErange	Determina o tamanho da área de busca na predição inter-quadros.	0...32768
Ref	Limita o número máximo de referências L0.	0...16
SubME	Define o nível de refinamento de sub-pixeis aplicado na predição inter-quadros.	0...7

previamente codificados, como por exemplo o primeiro quadro de um vídeo ou de um GOP (*Group Of Pictures*) ou um quadro RID. O segundo tipo são os quadros-P, que podem ser codificados com a predição inter-quadros utilizando como referência quadros temporalmente antecedentes já codificados. Desta forma, quadros-P possuem maior custo e melhor eficiência compressão do que os quadros-I. Por fim tem-se os quadros-B, que em virtude de permitirem o uso de referências temporalmente anteriores e posteriores a ele, acabam tendo maior custo computacional e atingindo uma melhor eficiência de codificação do que os demais tipos. Os parâmetros que permitem alterar como estes tipos são aplicados aos quadros durante a codificação estão evidenciados na Tabela 4.

Tabela 4 – Parâmetros Relacionados aos Tipos de Quadros

Parâmetro	Função	Valores Assumidos
B-adapt	Define o nível de esforço para determinar a ordem dos quadros-B.	0   1   2
B-frames	Limita o número máximo de quadros-B consecutivos.	0...16
B-intra	Permite o uso da predição intra para quadros-B.	On/Off
RC-lookahead	Número de quadros que serão avaliados para melhorar as decisões de Scene-cut.	B-frames...250
Scene-cut	Responsável por determinar a latência na inserção de quadros-I.	On/Off
Weight-B	Ativa/Desativa o uso de predição ponderada em quadros-B.	On/Off
Weight-P	Ativa/Desativa o uso de predição ponderada em quadros-B.	On/Off

Por fim, o codificador ainda utiliza filtros de de-blocagem nos quadros reconstruídos para eliminar artefatos visuais deixados pela codificação. Neste processo o algoritmo de filtro *in-loop Sample Adaptive Offset* (SAO) pode ser utilizado ou desabilitado através do parâmetro "SAO".

## 2.3 Controle de Energia em Codificadores HEVC

Devido à alta complexidade e às muitas ferramentas utilizadas no HEVC, este padrão possui um alto custo de processamento e muitos parâmetros de configuração, como demonstrado no capítulo 2. Estes parâmetros podem ser modificados com a finalidade de alterar o funcionamento de determinados aspectos e ferramentas do codificador. Vale também salientar que estes parâmetros podem ser modificados internamente no codificador a qualquer momento da codificação.

Estas características dos parâmetros nos permitem alterar certos aspectos da codificação conforme desejado, dentre estes aspectos podemos citar o custo energético,

temporal e até mesmo aritmético do codificador. Porém, ao se manipular estes aspectos será causado um impacto na eficiência da codificação. Frente a isso, é possível utilizar tais parâmetros para controlar o comportamento do processo de codificação de forma dinâmica, balanceando a eficiência e o custo da codificação de acordo com a demanda do cenário. Todavia, para realizar este controle com eficácia, é necessário a utilização de um Sistema de Controle de Malha Fechada (SCMF) (OGATA; YANG, 2002), que monitore intermitentemente o estado da aplicação sendo controlada e atue sempre que for necessário alterar o comportamento desta aplicação.

De acordo com (OGATA; YANG, 2002), um sistema de controle em *loop* pode possuir cinco componentes que seguem o fluxo do diagrama de blocos apresentado na Figura 2. Neste diagrama o bloco (I) Instalação representa o processo que se deseja controlar (neste caso um codificador HEVC), sendo que seu comportamento é monitorado pelo (II) Sensor. Este Sensor observa uma ou mais variáveis de (I), e envia os dados obtidos para o (III) Detector do Erro, que por sua vez calcula o quanto a (I) Instalação está à cima ou à baixo do comportamento desejado. Em seguida, o erro calculado é enviado ao (IV) Amplificador para este ajustar o erro e os pontos de controle que devem ser utilizados pelo (V) Atuador, para este adequar o consumo da (I) Instalação para a próxima iteração. Este processo é então repetido até a (I) Instalação terminar o seu funcionamento.

O modelo de SCMF apresentado em (OGATA; YANG, 2002) é diretamente aplicado nos controladores desenvolvidos para o padrão HEVC por (CORREA et al., 2015) (GRELLERT, 2014), (PENNY, 2017) e (MACHADO, 2017). Com estes trabalhos alterando o funcionamento interno de alguns módulos para adaptar o modelo ao contexto de codificação de vídeo. Esta adaptação pode ser observada na Figura 3 que apresenta um diagrama de blocos genérico dos sistemas de controle utilizados nos controles desenvolvidos para codificadores HEVC até então.

Analisando o diagrama percebe-se que o bloco (I) Codificador HEVC irá codificar um vídeo e ter suas informações analisadas por um (II) Sensor, assim como ocorre nos blocos (I) e (II) do diagrama anterior. Em seguida, tem-se o (III) Bloco do Con-

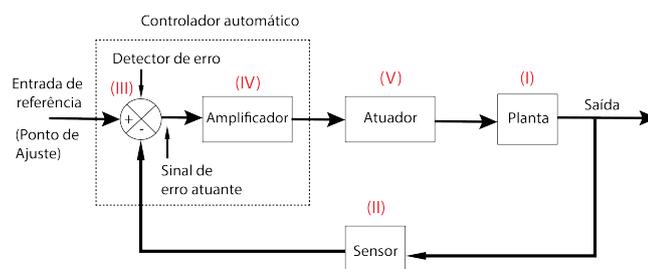


Figura 2 – Diagrama de Blocos do Sistema de Controle de Malha Fechada definido por (OGATA; YANG, 2002)  
(Fonte: Adaptado de OGATA; YANG (2002))

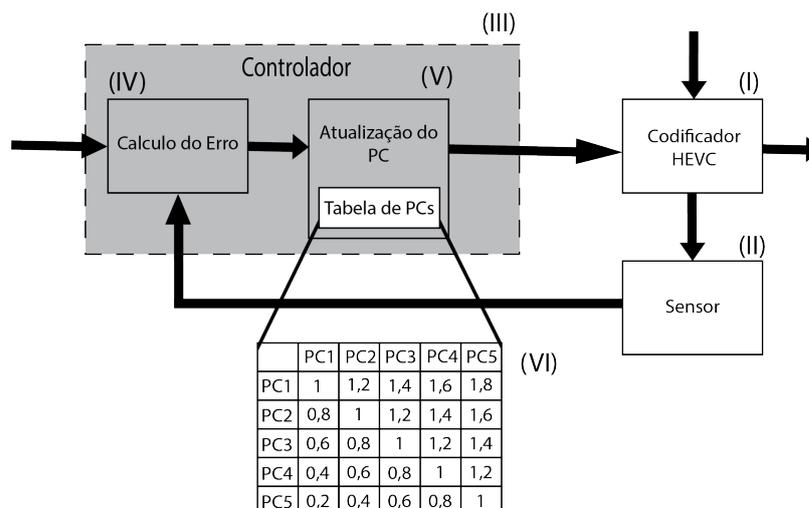


Figura 3 – Diagrama de Blocos do Sistema de Controle em codificadores HEVC

trolador, que geralmente é um algoritmo sendo executado em paralelo ao codificador e ao sensor. O controlador recebe como entrada do usuário, um *SetPoint* (SP), que diz quanto é esperado que se diminua o consumo de determinada variável no processo de codificação. Dentro do controle tem-se dois blocos de operação, sendo o (IV) Cálculo do Erro um bloco equivalente a uma junção dos blocos Detector do Erro e Amplificador de OGATA; YANG (2002). Este bloco é responsável por determinar o erro entre o consumo do codificador e o SP a ser atingido e informá-lo com precisão ao bloco de (V) Atualização do PC, por sua vez análogo ao bloco Atuador. Por fim, com as informações recebidas o Atuador busca em uma (VI) Tabela de Pontos de Controle (TPCs) qual a próxima configuração a ser utilizada e aplica esta ao codificador. Esta tabela contém de forma ordenada a distância normalizada do consumo de todos os PCs em relação ao atualmente ativo. Desta forma o atuador apenas percorre a coluna referente ao PC atual até chegar no fator de redução mais próximo do erro calculado. Vale também ressaltar que o período de atuação pode mudar de controlador para controlador. O período de atuação pode variar desde o processamento de uma CTU até o de um ou mais GOPs.

### 2.3.1 Trabalhos Relacionados

Existem muitas semelhanças e diferenças entre as metodologias utilizadas nos controles para codificadores HEVC presentes na literatura.

GRELLERT (2014) desenvolve um algoritmo para o codificador HM que visa controlar o tempo de codificação através da redução do número de operações aritméticas realizadas pelo codificador. Para isso o autor seguiu um fluxo similar ao do diagrama apresentado na Figura 3 sendo que o controlador atua em nível de CTU e utiliza como sensor uma leitura sobre a complexidade aritmética (CA) de cada CTU codificada do quadro atual. Ao fim da codificação de um quadro, as CAs de todas as suas CTUs

Tabela 5 – Tabela de Parâmetros Avaliados por GRELLERT (2014)

	c0	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12
<b>AMP</b>	1	0	1	1	1	1	1	1	1	1	1	1	1
<b>FME</b>	1	1	0	1	1	1	1	1	1	1	1	1	1
<b>Had. ME</b>	1	1	1	0	1	1	1	1	1	1	1	1	1
<b>Max. CTUd</b>	4	4	4	4	1	2	3	4	4	4	4	4	3
<b>Max. RF 4</b>	4	4	4	4	4	4	4	1	4	4	4	4	3
<b>SR</b>	64	64	64	64	64	64	64	64	16	32	8	64	64
<b>Max. TUD</b>	3	3	3	3	3	3	3	3	3	3	3	1	2

são somadas para gerar a complexidade total demandada pelo quadro. Em seguida, a CA obtida pela soma é subtraída do SP para assim gerar o Erro em relação à complexidade esperada.

Após o cálculo deste erro, o controle utiliza um sistema de *budget* para decidir quais PCs utilizar em cada CTU sendo codificada. Neste sistema foi implementado um controle de PID (*Proportional Integral Derivative*) (WILLIS, 1999) para efetuar as decisões. Um controle PID utiliza componentes proporcional, derivada e integral em laço fechado para determinar uma atuação diante de um determinado erro observado. Para operar, o controle (atuador) utiliza o erro para selecionar uma configuração em uma tabela denominada CFG-LUT (*Configuration Lookup Table* - Tabela de PCs) que contém todos os PCs do controlador. Com a finalidade de tornar esta decisão mais eficiente, o atuador prioriza a redução de CA em CTUs menos custosas utilizando informações sobre a profundidade utilizada em cada CTU no último quadro, sendo estas armazenadas em uma tabela nomeada de CTU-HT (CTU History Table).

A formação dos PCs integrantes da tabela CFG-LUT e a determinação de suas reduções foi realizada através de uma análise de sensibilidade constituída por múltiplas etapas e experimentos. Primeiramente o autor avaliou isoladamente 7 dos principais parâmetros de codificação, totalizando 12 configurações do codificador HM apresentadas na tabela 5 juntamente com a configuração padrão dos parâmetros (C0). Em seguida, as informações obtidas foram utilizadas para formar 4 novos PCs com consumos estimados, para desta forma prover heterogeneidade dos PCs do controle. Em outras palavras, se na etapa inicial constatou-se que o parâmetro X reduziu CA em 5% e o parâmetro Y em 10%, assumiu-se que uma configuração utilizando X e Y reduziria a CA em 15%. A redução de CA de cada PC foi estimada em 20%, 40%, 60% e 80%, com estes sendo respectivamente nomeados de PS20, PS40, PS60 e PS80. Contudo, segundo o autor, os PCs tiveram que ser refeitos duas vezes, pois a interinfluência dos parâmetros tornou imprecisa a estimativa das reduções dos PSs. Após o término da análise de sensibilidade, quatro PCs com reduções de 17%, 38%, 68% e 80% foram selecionadas para a tabela de PCs (CFG-LUT) do controle, apresentando assim um erro de até 8% quando comparado com a redução desejada originalmente.

Já o controlador proposto por CORREA et al. (2015) foi desenvolvido para o codificador HM (versão 13.0) e também segue uma estrutura em *loop* similar à de OGATA; YANG (2002), porém foca em redução de tempo de codificação e utiliza técnicas para efetuar atuações de controle em dois níveis de granularidade, possuindo assim dois atuadores. Para isto, o tempo de codificação de cada GOP é monitorado através de um sensor e transmitido para um bloco de Computação do Fator de Redução de Complexidade (CRFC). O CRFC então calcula uma média ponderada entre o custo temporal dos dois últimos GOPs codificados e determina erro entre o consumo esperado (SP) e o real. Este erro então serve como um fator de ajuste que indica o quanto deve-se aumentar ou diminuir o consumo temporal do codificador no próximo GOP. Tendo este fator calculado, o controlador entra em um bloco atuador chamado *Medium Granularity Time Control* (MGTC) ou, Controlador de Tempo de Granularidade Média, que modifica a configuração utilizada pelo codificador buscando em uma tabela de PCs com 15 configurações a configuração com o fator de ajuste mais próximo do calculado para ser utilizada no próximo GOP.

Ainda, este trabalho possui outra grande diferença metodológica em relação ao de GRELLERT (2014) na análise de sensibilidade. Para selecionar os pontos de controle o autor realizou uma avaliação RDC (*Rate Distortion Complexity*) para tentar encontrar as configurações com melhores compromissos entre compressão, qualidade de imagem e custo temporal. Nesta avaliação foram utilizados os parâmetros de *Search Range* (Janela de busca - SR), *Hadamard Motion Estimation* (HME), *Bi-Prediction* e três esquemas de terminação antecipada propostas em CORREA et al. (2015), totalizando 240 configurações analisadas. Após coletar as informações R-D-C de todas as configurações, foi utilizada a técnica de Ótimo de Pareto para identificar as configurações mais eficientes, cujo conjunto destas é denominado de Frente de Pareto. Tendo em vista que a técnica de Pareto serve para avaliar apenas duas variáveis simultaneamente, foi necessário aplicar a técnica primeiramente considerando os dados de R-C e em seguida de D-C. Tendo ambas frentes identificadas, foram selecionadas apenas as configurações presentes em ambas seleções, para desta forma garantir que apenas as configurações que possuam a melhor eficiência nas três variáveis fossem utilizadas como PC. Apesar da utilização do Pareto permitir uma análise mais ampla de um maior número de configurações, o seu funcionamento não permite que o desenvolvedor do controle determine a heterogeneidade das reduções de cada PC. Frente a esta deficiência da técnica, foram selecionados PCs adjacentes na tabela de PCs cujas diferenças de redução de complexidade variavam de 1,5% a 21%, mostrando limitada heterogeneidade das reduções dos PCs.

Devido a pouca heterogeneidade das reduções dos PCs escolhidos, foi observado que o controle desenvolvido muitas vezes ficava alternando entre dois PCs sem conseguir se aproximar do SP, por este estar entre a redução provida pelos dois PCs

adjacentes na TPCs. Para lidar com estes casos, CORRÊA et al. (2012) criou um novo bloco de controle atuador denominado *Fine Granularity Time Control* (Controle Temporal de Granularidade Fina) (FGTC) responsável por efetuar pequenas reduções de complexidade não cobertas pelos PCs. O FGTC também recebe como entrada do CRFC o fator de ajuste, porém o utiliza em conjunto com informações sobre a profundidade das CTUs vizinhas para restringir a profundidade máxima da CTU sendo codificada e limitar o número de PUs que podem usar tamanhos menores que  $2N \times 2N$ . Tomando estas decisões em nível de CTU, o controle consegue providenciar as reduções de tempo não abrangidas pelos PCs.

Na literatura ainda existem os controladores de PENNY (2017) e MACHADO (2017) que focam em realizar controle energético em codificadores HEVC. Ambos seguem uma metodologia muito similar à de CORREA et al. (2015), contendo apenas pequenas modificações metodológicas e funcionais descritas a seguir. Embora PENNY (2017) realize controle no codificador HM (versão 15.2) e MACHADO (2017) no codificador x265 (versão 2.5), ambos controladores utilizam a ferramenta RAPL (*Running Average Powerlimit*) (PAN, 2013) como sensor para monitorar o consumo energético do processo de codificação e ajustá-lo a cada GOP processado da mesma maneira que o MGTC de CORREA et al. (2015). O RAPL é uma ferramenta desenvolvida pela Intel que permite a leitura de dados sobre o consumo energético de diversos componentes de um computador através de registradores MSR (*Model-Specific Registers*) presentes nas arquiteturas Intel Ivy e Sandy Bridge (PAN, 2013). Já as análises de sensibilidade efetuadas em ambos trabalhos possuem diferenças consideráveis.

Primeiramente PENNY (2017) utiliza cinco vídeos distintos para avaliar a eficiência energética de 6 parâmetros, totalizando 11 configurações. Em seguida, de forma similar à GRELLERT (2014), 31 configurações foram criadas com consumos estimados a partir de diferentes uniões das possíveis configurações dos 6 parâmetros avaliados. Por fim, assim como em CORREA et al. (2015) a técnica de Ótimo de Pareto foi utilizada para selecionar as melhores configurações, porém considerando os seus compromissos em relação ao R-D-E (*Rate-Distortion-Energy*). Embora a técnica tenha selecionado mais do que 10 configurações, as que apresentavam eficiências e consumos muito próximos foram ignoradas para evitar uma heterogeneidade muito baixa entre o consumo dos PCs. Seguindo este método de seleção, os PCs adjacentes na tabela de configurações utilizada no controlador acabaram tendo uma diferença entre seus consumos que varia de 3% até 13%, disformidade que pode causar inconsistências no comportamento do controle.

Já MACHADO (2017) utiliza apenas um vídeo na sua análise de sensibilidade para avaliar diversos parâmetros do codificador x265. Esta avaliação resultou em 48 configurações iniciais onde cada uma alterava apenas um parâmetro. Em seguida, da mesma forma que em PENNY (2017), estas configurações iniciais com parâmetros

isolados foram mescladas para gerar 86 novas configurações, onde as oito configurações que apresentaram melhor eficiência RDE foram selecionadas pela técnica de Pareto e, conseqüentemente, utilizadas na tabela de PCs do controlador que está representada na Tabela 6.

Tabela 6 – Tabela de PCs do Controlador de MACHADO (2017)

PCs	PC0	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
PC0	1,000	1,777	1,857	1,865	2,135	2,254	2,519	2,671	3,079
PC1	0,563	1,000	1,045	1,049	1,202	1,268	1,417	1,503	1,733
PC2	0,539	0,957	1,000	1,004	1,150	1,214	1,356	1,438	1,658
PC3	0,536	0,953	0,996	1,000	1,145	1,209	1,351	1,432	1,651
PC4	0,468	0,832	0,870	0,873	1,000	1,056	1,179	1,251	1,442
PC5	0,444	0,788	0,824	0,827	0,947	1,000	1,117	1,185	1,366
PC6	0,397	0,706	0,737	0,740	0,848	0,895	1,000	1,060	1,223
PC7	0,374	0,665	0,695	0,698	0,800	0,844	0,943	1,000	1,153
PC8	0,325	0,577	0,603	0,606	0,693	0,732	0,818	0,867	1,000

Devido ao maior número de parâmetros avaliados no x265, a heterogeneidade da redução dos PCs selecionados pelo Pareto acabou se mostrando ainda mais disforme. Como pode ser observado, alguns PCs adjacentes na TPCs possuem diferenças nos seus consumos que vão desde 0,4% (PC2 e PC3) até 44% (PC0 e PC1). Tal disparidade pode levar o controlador a possuir uma considerável faixa de redução energética inatingível. Por exemplo, caso uma codificação começando no PC0 precise reduzir em 30% o seu consumo, a configuração mais próxima dessa redução seria a PC1, que por consumir 44% a menos do que o PC0 faria a codificação economizar 14% a mais do que o desejado e, conseqüentemente, causando perdas desnecessárias na eficiência da codificação.

De maneira geral podemos salientar que as principais diferenças metodológicas entre os trabalhos de PENNY (2017) e MACHADO (2017) é que enquanto PENNY (2017) avalia 31 configurações com um conjunto de seis parâmetros para cinco vídeos no codificador de referência HM, MACHADO (2017) avalia 86 configurações com um conjunto de 22 parâmetros para um vídeo no codificador comercial x265. Já os métodos de seleção por Pareto e implementação do controle são similares.

### 2.3.2 Desafio de Pesquisa

Conforme discutido na seção anterior, as atuais metodologias de seleção de pontos de controle para controladores em codificadores HEVC são realizadas empiricamente por um especialista. Isto faz com que o desenvolvimento do controle se torne lento e impossibilita a avaliação de um número aceitável de configurações. Adicionalmente, algumas metodologias utilizam Pareto para selecionar os pontos ótimos, sendo que os

resultados desta técnica é completamente dependente do conjunto de entrada. Além disto, a seleção por Pareto ainda não permite com que o desenvolvedor selecione a heterogeneidade do consumo dos seus pontos de controle, que é fundamental para a performance do controlador.

Por fim, como as configurações são manualmente montadas por um especialista e os parâmetros que as compõem possuem alta interdependência, existe uma considerável chance de falha humana onde o especialista não conseguirá estimar corretamente a eficiência e redução da configuração. Desta forma, torna-se necessário o desenvolvimento de algum método de se encontrar configurações eficientes de forma autônoma e que consiga avaliar um conjunto considerável de configurações para garantir que os PCs encontrados sejam o mais eficientes possível.

Uma técnica que pode ser aplicada para resolver este problema de encontrar configurações eficientes é a aplicação de Algoritmos Genéticos. Estes algoritmos são capazes de automaticamente avaliar milhares de configurações possíveis e de atingir configurações cada vez mais eficientes através de várias iterações. Ainda, estes algoritmos seriam independentes de contexto e de especialista, excluindo riscos de falha humana enquanto ainda permitiria o desenvolvedor a selecionar com precisão a heterogeneidade desejada para os PCs do seu controlador. Algoritmos genéticos são uma classe de soluções focadas para solucionar problemas de otimização multiobjetivo. Estes problemas multiobjetivos e as classes de soluções para resolve-los serão mais profundamente discutidos no Capítulo 3.

### 3 PROBLEMAS DE OTIMIZAÇÃO MULTIOBJETIVO E ALGORITMOS GENÉTICOS

Um problema de otimização (PO) é definido pela busca da melhor solução dentro de um conjunto de soluções possíveis, onde a melhor solução minimiza (ou maximiza) uma função objetivo dentro de uma série de restrições. Esta classe de problemas pode ser dividida em duas categorias que dependem se suas variáveis são contínuas ou discretas. Um problema de otimização contínuo é modelado em cima de números reais que estão geralmente presentes em áreas como Física, Química, Engenharia, Estatística e Economia. Já um problema de otimização com variáveis discretas pode ser chamado de otimização combinatória e o seu objetivo geralmente é encontrar um inteiro, uma permutação ou um grafo dentro de um conjunto finito de soluções.

Na maioria dos casos reais é necessário a otimização de múltiplas variáveis onde, de forma geral, aperfeiçoar uma destas implica em perdas nas demais variáveis. Nestes casos o problema acaba sendo denominado de problema de otimização multiobjetivo (POMO), devido a necessidade de otimizar múltiplas variáveis. A interdependência das variáveis nos POMOs tornam estes problemas consideravelmente complexos de serem resolvidos, o que torna impossível a tarefa de se encontrar o melhor resultado na maioria dos casos.

A técnica de Frente de Pareto é empregada para selecionar pontos ótimos em problemas multiobjetivos. Esta simples seleção originária da economia aponta os pontos não dominados entre um conjunto de soluções levando em consideração dois eixos. Um ponto não dominado é aquele que para um valor no eixo X, não existe nenhum outro ponto no conjunto avaliado com um melhor valor no eixo Y. A Figura 4 apresenta um exemplo de seleção de Pareto entre as variáveis de Tempo e Espaço para um conjunto de 5 pontos onde uma linha representando a Frente de Pareto selecionada passa pelos pontos não dominados p, q e r.

Contudo, esta técnica possui algumas fraquezas em sua seleção, como por exemplo só consegue avaliar dois eixos e sua eficiência é completamente dependente do conjunto de pontos inserido. Em virtude destes pontos fracos e da complexidade de se solucionar problemas multiobjetivos, surgiram diversas classes de algoritmos que ten-

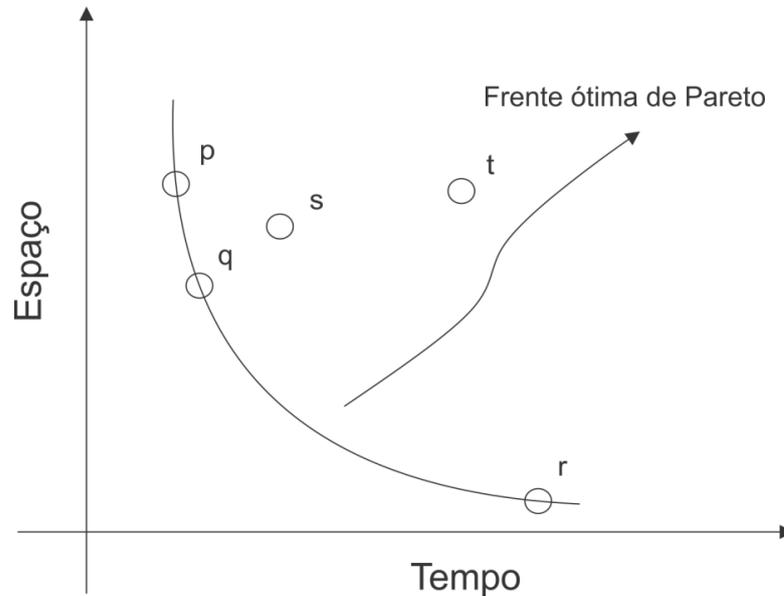


Figura 4 – Seleção Via Frente de Pareto  
(Fonte: Adaptado de GOSH (2004))

tam buscar resultados sub-ótimos (VIKHAR, 2016). Entre estas classes podemos citar os métodos de busca aleatória guiada, também denominados de métodos estocásticos, que são largamente utilizados graças as suas altas robustezas e performances.

Estas características apresentadas pelos métodos estocásticos se devem ao fato destes se basearem em regras de probabilidade, o que possibilita com que a busca saia de máximos locais e localize os mínimos globais. Entre os algoritmos estocásticos mais utilizados está a Tempura Simulada, porém, este algoritmo busca apenas um ponto por vez, o que limita a sua eficiência. Já os algoritmos evolucionários trabalham com uma população de pontos, e conseqüentemente, conseguem avaliar vários picos das melhores soluções simultaneamente, o que aumenta sua eficiência e chance de encontrar o mínimo global. Dentro da classe de algoritmos evolucionários, os AGs tendem a encontrar a melhor solução global com uma boa frequência (SOARES, 1997). Além disto, é uma das técnicas mais consolidadas e bem utilizadas dentre os algoritmos evolucionários (ABD-ALSABOUR, 2014) e adequados para problemas complexos sem solução conhecida (TABASSUM; MATHEW et al., 2014).

Os AGs pertencem à classe de Algoritmos Evolucionários (AEs) que simulam o processo de evolução natural descrito por DARWIN (1859). Desta forma, estes algoritmos mapeiam características e fenômenos genéticos que ocorrem em populações de espécies para o seu funcionamento. Ademais, um AG desenvolvido para resolver um problema de otimização multiobjetivo é denominado de Algoritmo Genético Multiobjetivo (AGMO) por seus operadores necessitarem otimizar múltiplos eixos interdependentes de maneira simultânea.

Para entendermos melhor os conceitos genéticos mapeados e o funcionamento de

um AG, vamos considerar uma determinada alcateia de lobos. Esta alcateia representa uma população de diversos indivíduos da espécie lobo onde cada lobo possui diversos genes no seu código genético. A combinação destes diversos genes ditam as características deste lobo, que são denominadas de fenótipos e, conseqüentemente, tornam o lobo mais apto ou não a sobreviver. Ainda, estes lobos irão se reproduzir para formar novas gerações desta população (alcateia), sendo que o novo indivíduo irá conter uma mistura entre os genes do pai e da mãe. Nestas reproduções os lobos tendem a selecionar, na maioria das vezes, os parceiros mais aptos para a reprodução, garantindo uma maior probabilidade de que a próxima geração seja mais evoluída (apta) do que a atual. Entretanto, em qualquer população é necessário que os menos aptos também se reproduzam, para assim evitar com que a espécie se torne muito parecida e acabe tendo problemas genéticos. Outro fenômeno genético importante e extremamente influente na evolução de espécies é a mutação, onde o código genético de um novo indivíduo sofre uma alteração aleatória. Este fenômeno viabiliza a injeção de novo código genético na população, pois pode inserir novas características em indivíduos e torná-los mais ou menos aptos. Por fim, em uma alcateia, o lobo alfa nunca é banido e desta forma sempre participa da próxima geração, garantindo assim que através de um elitismo os genes mais evoluídos estejam presentes na próxima geração.

Um algoritmo genético se comporta e possui os mesmos fenômenos genéticos da alcateia de lobos descrita, com estes fenômenos recebendo o nome de operadores genéticos. No primeiro passo de um AG, o algoritmo genético gera uma população inicial de indivíduos de forma geralmente aleatória. Estes indivíduos por sua vez são formados por vários genes especificados pelo problema sendo abordado, podendo ser por exemplo uma cadeia de caracteres ou variáveis sobre tamanho, consumo, complexidade, qualidade, etc. Estes genes vão obrigatoriamente influenciar nos fenótipos dos indivíduos da população, os tornando mais ou menos aptos. O nível de aptidão de um indivíduo é determinado no AG por uma função de avaliação denominada *Fitness* que deve relacionar as variáveis sendo avaliadas pelo algoritmo caso a otimização seja multiobjetivo. Esta função é então utilizada como critério na etapa de seleção, onde são determinados os pares de lobos que vão se reproduzir. Nesta seleção é importante dar-se maior probabilidade de indivíduos mais aptos se reproduzirem para assim garantir que a próxima geração seja mais evoluída que a atual, e conseqüentemente, levar o algoritmo a convergir para uma solução sub-ótima. Tal probabilidade é denominada de pressão de seleção, e caso seja muito alta pode causar uma convergência muito rápida no algoritmo, conseqüentemente podendo trancá-lo em mínimos locais.

Como afirmado anteriormente, a evolução através das populações acontece através da troca de genes entre os dois pais realizando a reprodução. Em um AG esta

troca ocorre em todas as reproduções dos indivíduos e é denominada de *Crossover*, com exceção de quando as operações de elitismo e mutação ocorrem. O operador de elitismo em um AG ocorre quando o indivíduo mais apto (análogo ao lobo Alfa) é passado sem modificação alguma para a próxima geração. Desta forma se garante que a melhor solução encontrada até então seja preservada e também a evolução da população. Já a mutação ocorre em um indivíduo de forma aleatória para inserir novo material genético na população e conseqüentemente possibilitar com que esta possa escapar de um mínimo local.

A Figura 5 apresenta o fluxograma do funcionamento de um AG. Onde inicialmente uma população é criada aleatoriamente e cada indivíduo tem seu valor de *fitness* calculado. Caso o critério de parada tenha sido encontrado, o melhor indivíduo da última população é retornado, caso contrário, a população será submetida aos operadores genéticos. Nesta etapa os indivíduos da população são submetidos aos processos de seleção, *crossover* e mutação, para formar uma nova geração de indivíduos. Então esta nova geração também é levada ao passo de ter seus valores de *fitness* calculados, completando o laço do algoritmo genético. Devido à diversidade de contextos onde AGs e AGMOs são aplicados, existem diversas propostas de como realizar os operadores genéticos. Sendo assim, no capítulo seguinte serão discutidas cada um dos operadores apresentados juntamente com as maneiras de aplicá-los.

### 3.1 Seleção

Dentre os métodos mais comuns de realizar a seleção dos indivíduos para reproduzir estão a Roleta, a Seleção por Classificação e o Torneio. A técnica de roleta foi definida no primeiro SGA (*Simple Genetic Algorithm*) proposta por HOLLAND et al. (1992) e consiste em somar os valores de *fitness* de todos os indivíduos de uma população e, em seguida, sortear um número aleatório entre um e o total somado. Tendo o número definido, a seleção deve percorrer os indivíduos da população enquanto diminui os seus valores de *fitness* do valor sorteado. Por fim, o indivíduo cujo *fitness* tornar o valor sorteado menor ou igual a zero será selecionado para participar da próxima geração ou realizar *crossover*. Desta forma, os indivíduos mais aptos (com maior *fitness*) ocupam uma maior porção da roleta de forma proporcional ao seu valor de *fitness* e, conseqüentemente, tem uma maior chance de serem selecionados. Esta distribuição de probabilidade pode ser observada na Figura 6 que apresenta uma roleta com cinco indivíduos juntamente com os seus valores de *fitness*, neste caso também suas probabilidades de seleção, que estão exibidos nas suas respectivas distribuições.

Em cenários onde os valores de *fitness* dos indivíduos possuam grande variação (alto desvio padrão), a roleta pode dar chances ínfimas de seleção para indivíduos com aptidões muito baixas devido a distância do seu *fitness* com o dos mais aptos.

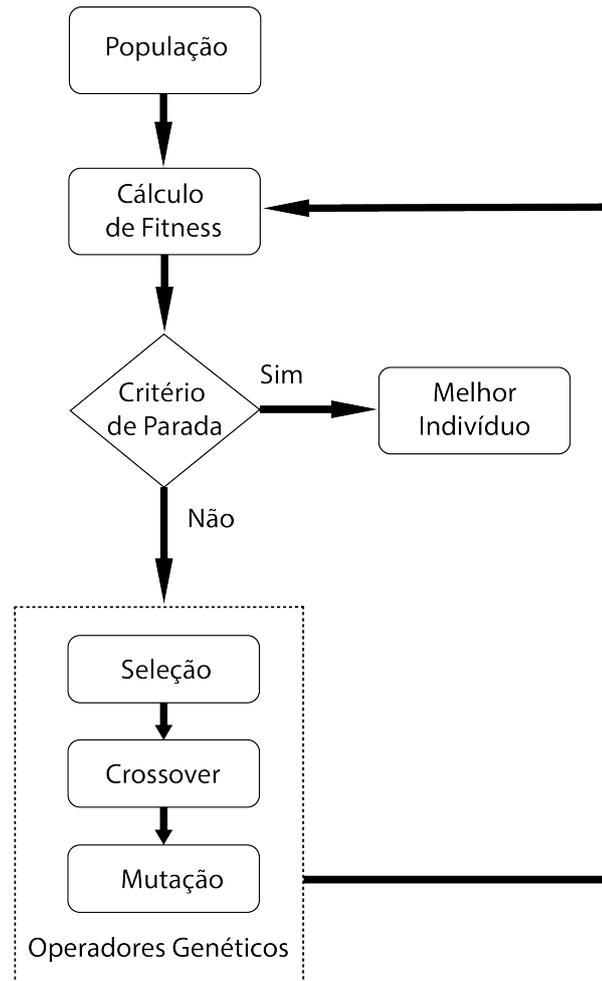


Figura 5 – Fluxograma de um Algoritmo Genético

De forma geral esta situação é indesejável frente que o método pode causar uma pressão de seleção muito alta, mantendo apenas os genes mais aptos e aumentando a chance do algoritmo ficar preso em mínimos locais devido a rápida convergência. Para adaptar este método de seleção a tais cenários, foi proposta a Seleção por Classificação. Neste método os indivíduos são organizados de acordo com seus valores de *fitness* do maior ao menor. Em seguida, um valor crescente é atribuído aos indivíduos, com o de pior aptidão recebendo o valor 1 e o indivíduo mais apto o valor N, sendo N o número de indivíduos presentes na população. Com estes novos valores, os indivíduos mais aptos ainda possuem uma chance maior de serem escolhidos, enquanto ainda são factíveis as chances de seleção daqueles que possuem valores de *fitness* muito baixos. Embora tal método solucione o problema de grandes variações nos *fitness* de uma população, ele pode causar uma menor convergência, já que os melhores cromossomos acabam se distinguindo menos dos demais. Por fim, a seleção por Torneio consiste em escolher aleatoriamente um subconjunto da população e selecionar o que tiver o melhor valor de *Fitness* (GOLDBERG, 1989). A operação também pode ser efetuada duas vezes para selecionar dois indivíduos distintos para

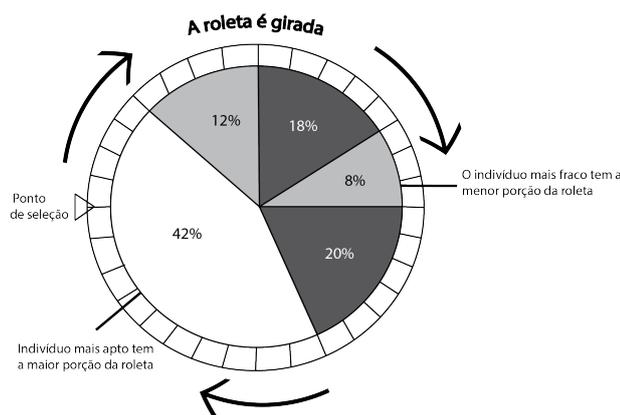


Figura 6 – Distribuição estatística da seleção de indivíduos no método de roleta (Fonte: Adaptado de SUKSTRIENWONG (2018))

se reproduzir e realizarem o *crossover*.

Apesar dos métodos de seleção descritos serem largamente utilizados e funcionais, eles não se aplicam adequadamente em algoritmos genéticos multiobjetivo em vista de que eles avaliam apenas um valor de *fitness* ao invés de dois ou mais. Neste contexto são propostas técnicas mais avançadas onde é comum a utilização do conceito de Ótimo de Pareto como técnica auxiliar para definir uma classificação para os indivíduos. Entre os métodos de seleção para cenários multiobjetivos encontra-se o trabalho de GOLDBERG (1989), que propõe uma aplicação de valores de classificação juntamente com a técnica de Ótimo de Pareto. Neste método todos os indivíduos não dominados na frente de Pareto recebem a classificação 1 e, logo, possuem a mesma chance de reprodução. Após ranqueados, os pontos são excluídos da seleção e um novo conjunto de indivíduos não dominados é selecionado e todos indivíduos são atribuídos à classificação 2. Assim o processo é repetido sucessivamente até todos os indivíduos possuírem uma classificação. Por fim, um método como o de roleta pode ser aplicado com base na classificação dos indivíduos.

Na literatura ainda consta o método de FONSECA; FLEMING (1998), que distribui os indivíduos de uma população em uma distribuição de Pareto e atribui uma classificação a cada indivíduo de acordo com o número de indivíduos que os dominam. Sendo assim, os indivíduos não dominados (mais aptos) recebem classificação 0, os que são dominados por 1 outro indivíduo classificação 1, por 2 indivíduos classificação 2 e assim por diante. Vale ressaltar que ao utilizar tal método a seleção deve considerar que os valores de classificação mais baixos são os de maior aptidão.

Por fim HORN; NAFPLIOTIS; GOLDBERG (1994) propõe um método denominado *Niched Pareto*. Nesta seleção dois indivíduos são aleatoriamente escolhidos e então uma seleção por torneio entre os dois indivíduos é realizada. Neste método ambos indivíduos são comparados com uma sub-população distribuída ao seu redor, ou seja, seu nicho. O indivíduo que for menos dominado por membros do seu nicho acaba

sendo selecionado. Ainda, caso ambos não sejam dominados por nenhum outro indivíduo ou possuam mesmo grau de dominação, então diferentes métodos de *fitness sharing* são aplicados de acordo com o contexto para assim manter a heterogeneidade na distribuição do conjunto de soluções presentes no Pareto.

### 3.2 Crossover

A eficiência da operação genética de *crossover* é consideravelmente dependente do problema em que está sendo aplicada, possuindo assim diversas maneiras de ser realizada. Pode-se citar como as maneiras mais simples os *crossovers* de ponto único, multi-ponto e uniforme.

No *crossover* de ponto único determina-se de forma aleatória um ponto de *crossover* na cadeia genética de ambos pais. Em seguida, um ou dois indivíduos filhos são produzidos, com cada um possuindo os genes de ambos pais onde a origem de cada gene é definida pelo ponto de *crossover* determinado. Desta forma se um indivíduo possui 4 genes e o ponto de *crossover* recebeu o valor 2, o primeiro e o segundo gene do filho serão provenientes do primeiro pai enquanto o terceiro e o quarto genes serão herdados do segundo pai. A Figura 7 demonstra a criação de um filho a partir de uma operação de *crossover* de ponto único, com o ponto de divisão sendo representado pela seta vertical. Já a Figura 8 mostra o funcionamento do *crossover* de dois pontos, cuja única diferença é que dois pontos de *crossover* são definidos para demarcar as origens dos genes.

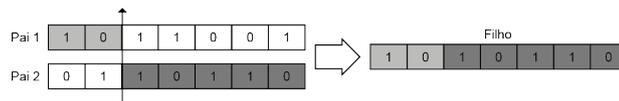


Figura 7 – *Crossover* de Ponto Único

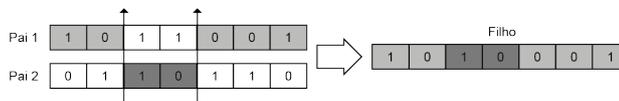


Figura 8 – *Crossover* de Dois Pontos

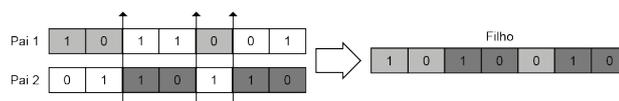


Figura 9 – *Crossover* Uniforme

Já no *crossover* uniforme demonstrado na Figura 9, cada gene do indivíduo filho sendo produzido tem sua origem definida aleatoriamente. Neste método as trocas são feitas gene a gene, não forçando a seleção de cadeias inteiras de genes de um dos

pais. Contudo, dependendo do contexto é possível colocar um peso para os genes de um dos pais terem uma maior probabilidade de serem transmitidos ao filho.

Ao se selecionar uma técnica de *crossover* para o AG é necessário observar que as técnicas podem causar um viés de proporção ou de distribuição dependendo do problema sendo modelado. Um viés de proporção é causado quando a seleção dos genes possui uma dependência com relação as suas posições, por exemplo, se em um *array* de *bits* os mais significativos vierem do pai 1, o filho terá uma tendência a se parecer mais com este pai do que com o outro. Já um viés de proporção está relacionado com a dependência em relação a quantidade de bits que será transferido de cada pai, ou seja, se considerarmos o mesmo *array* de *bits*, existe uma grande chance de um dos pais passar muito mais *bits* do que o outro, causando assim maior influência sobre o filho.

Frente a isto, podemos afirmar que entre os diversos métodos de *crossover*, a técnica de ponto único possui o maior viés de proporção, enquanto a técnica multi-ponto possui o maior viés de distribuição. Tais características fazem com que estas técnicas acabem influenciando negativamente a convergência do algoritmo, o que o leva a gerar populações excessivamente homogêneas e, conseqüentemente, a explorar um menor número de possíveis soluções devido à rápida convergência. Embora a técnica uniforme possua o menor viés dentre as discutidas, ela acaba perturbando consideravelmente a distribuição genética devido a sua natureza altamente aleatória porém, ainda assim estudos mostram que esta é técnica mais indicada para casos onde as populações possuem tamanho reduzido (SRINIVAS; PATNAIK, 1994).

Diversos cenários onde AGs são aplicados possuem genes representados por números reais onde realizar uma troca pura de genes pode não ser interessante. Em vista destes cenários, foram propostas técnicas como *Crossover* Intermediário (CI), Heurístico (CH) e Aritmético, também denominado de *Flat*, onde os genes de ambos os pais são misturados no filho através da aplicação de uma função matemática em seus genes. Na técnica intermediária é transmitido para os filhos uma média ponderada dos genes de ambos os pais definida pela equação 1. O peso aleatório representado por "*rand*" é multiplicado por uma "taxa" que pode estar no intervalo [0-1], ser um escalar ou ainda um vetor com um determinado número de variáveis (KAYA; UYAR et al., 2011). Já o *Crossover* Heurístico produz um filho mais próximo do pai com maior *fitness* (KAYA; UYAR et al., 2011). A função responsável por tal interação entre os genes dos pais, assumindo que o "Pai2" possua melhor aptidão, está representada na equação 2, onde a variável "taxa" responsável por determinar a proximidade do filho com o pai de maior *fitness* é inserida pelo desenvolvedor e normalmente recebe o valor de 1,2. Já o método Aritmético aplica uma média ponderada entre os genes dos dois pais dando uma prioridade para o pai com maior aptidão. A equação 3 demonstra um caso de *crossover* aritmético onde o "Pai1" possui maior *fitness* sendo que o peso

da ponderação representado por "rand" é um número aleatório no intervalo [0-1] que deve ser aplicado igualmente a todos os genes (MAGALHAES-MENDES, 2013).

$$Filho = Pai2 + rand * Taxa * (Pai1 - Pai2) \quad (1)$$

$$Filho = Pai2 + Taxa * (Pai1 - Pai2) \quad (2)$$

$$Filho = rand * Pai1 + (1 - rand) * Pai2 \quad (3)$$

### 3.3 Mutação

Assim como o *crossover*, o operador de mutação também é responsável por alterar o código genético de indivíduos em uma população. Este é um operador cuja sua aleatoriedade causa uma perturbação no código genético de um ou mais indivíduos para possibilitar com que a população escape de máximos locais e mínimos globais. De maneira geral, na mutação um indivíduo é aleatoriamente selecionado para ter um de seus genes modificados de forma também aleatória.

Todavia, a probabilidade de mutações ocorrerem deve ser baixa e depende do contexto, estando normalmente no intervalo de 0% a 10% pois, caso contrário, a busca do algoritmo pode se tornar apenas uma busca aleatória (SILVA, 2001). Ainda, de acordo com (OCHOA; HARVEY; BUXTON, 2000), as taxas ótimas de mutação tendem a aumentar conforme a pressão de seleção cresce.

Existem múltiplas formas de se aplicar mutação em uma população. Entre elas estão:

- **Boundary:** o gene recebe o maior ou menor valor possível dentro de uma faixa definida pelo usuário.
- **Não-Uniforme:** a probabilidade de mutação vai diminuindo conforme as gerações avançam. Esta técnica permite um ajuste mais fino para tornar a busca mais localizada quando próxima do resultado final.
- **Gaussiana:** uma gaussiana é criada para os valores possíveis do gene escolhido para mutação e um novo valor é aleatoriamente definido para o gene. Caso este valor esteja fora da borda superior ou inferior da gaussiana ele recebe o valor da respectiva borda.

## 4 RDE-AGMO: SELETOR DE CONFIGURAÇÕES EFICIENTES PARA CODIFICADORES

O algoritmo proposto neste trabalho tem como principal funcionalidade buscar configurações de codificadores HEVC que mantenham boas eficiências em relação a taxa de compressão e qualidade da imagem enquanto reduz o consumo de energia da codificação em um determinado nível. Desta forma o AGMO é capaz de selecionar eficientemente de maneira *off-line* configurações para serem utilizadas na Tabela de PCs de controladores de energia como representado na Figura 10. Ao realizar esta tarefa o algoritmo é capaz de substituir a análise de sensibilidade *off-line* necessária para o desenvolvimento de controladores de energia de forma mais ampla e automática do que as metodologias atualmente aplicadas enquanto ainda possibilita selecionar com precisão o fator de redução de cada configuração na Tabela de PCs e, conseqüentemente, sua heterogeneidade. Para atingir este objetivo foi realizada uma avaliação sobre as formas mais eficientes de se aplicar cada um dos operadores genéticos presentes no algoritmo.

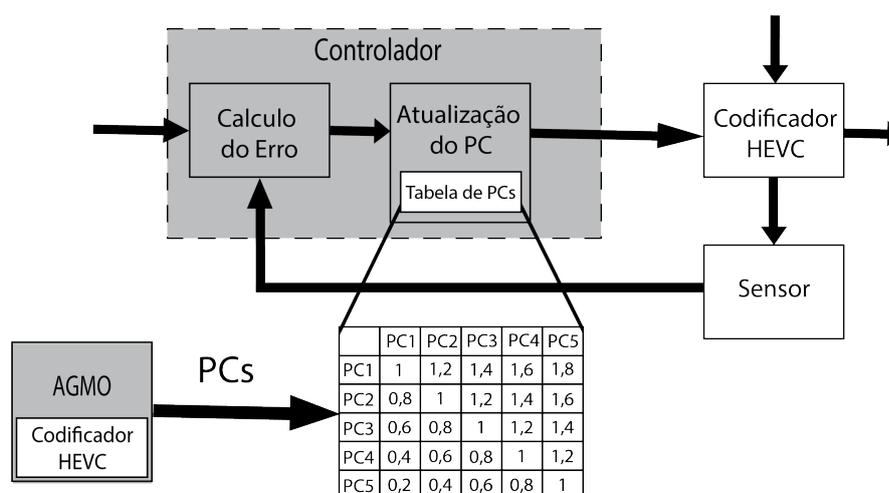


Figura 10 – Funcionalidade e Aplicação do RDE-AGMO

## 4.1 Modelagem e Definição dos Operadores Genéticos

O primeiro passo para se utilizar um AGMO em um dado problema de otimização é modelar este problema dentro dos conceitos genéticos da técnica. No caso abordado neste trabalho estamos buscando uma configuração sub-ótima do codificador HEVC que possua um bom compromisso entre a taxa de compressão e a qualidade de imagem enquanto mantém um determinado nível de redução no consumo energético da codificação. É importante também observar que uma configuração é composta por uma série de parâmetros que possuem influência no comportamento da codificação.

Em vista que um AGMO retorna uma população de indivíduos formados por um conjunto de genes que caracterizam as suas aptidões (eficiências), é natural estabelecermos que cada indivíduo é representado por uma configuração do codificador, sendo que cada parâmetro desta configuração caracteriza um gene do indivíduo. A Figura 11 demonstra a representação de um indivíduo no AGMO proposto utilizando uma configuração do codificador x265 e seus parâmetros. Na Figura 11, cada quadrado representa um gene, ou seja, o parâmetro seguido do seu atributo de configuração. Vale notar que ao total um indivíduo é constituído por 25 parâmetros de codificação.

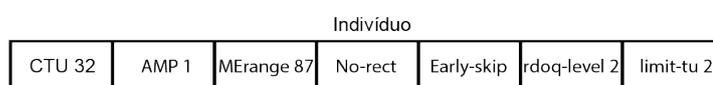


Figura 11 – Representação de um indivíduo

Para selecionar os melhores operadores genéticos para o algoritmo, é necessário levarmos em consideração como os genes dos indivíduos se comportam e influenciam os seus fenótipos, ou seja, como cada parâmetro das configurações funcionam e como eles impactam a eficiência das codificações. Também é muito importante conhecermos como a função de avaliação se comporta, neste caso, as características da codificação como tempo, possibilidade de paralelização, etc.

A função de avaliação do AGMO proposta é constituída pela aplicação das configurações na codificação de um vídeo utilizando o codificador x265, que por sua vez irá retornar as informações sobre compressão e qualidade de imagem enquanto a ferramenta RAPL coleta o seu consumo energético. Uma codificação demanda um tempo considerável para ser efetuada e além disso não pode ser paralelizado devido ao método sendo utilizado para coletar informações de energia. Tais características acabam limitando consideravelmente a possibilidade de utilizar populações grandes ou aumentar o número máximo de gerações sendo codificadas, pois isto implicaria em uma grande demanda de tempo, tornando os experimentos inviáveis.

Frente a estas dificuldades e o tempo tomado para realizar uma codificação, torna-se interessante a aplicação de técnicas que acabem fazendo o algoritmo ter uma con-

vergência mais rápida e possibilitem o uso de populações menores. Outro fator importante sobre a codificação é que o vídeo sendo comprimido possui grande influência no impacto dos parâmetros sendo aplicados. Tendo isto em mente, o trabalho utilizou o vídeo *PartyScene* para gerar os valores de *fitness*. Isto se deve pelo fato de que utilizar este vídeo torna mais justa as comparações com as configurações utilizadas no controle de MACHADO (2017), pois o mesmo vídeo foi utilizado na análise de sensibilidade deste controlador, que será utilizado como caso de estudo para avaliar o desempenho das configurações encontradas pelo AGMO.

Cada parâmetro de uma configuração no x265 possui uma faixa distinta de valores inteiros de entrada que podem variar desde 0 a 1 (Ligado/Desligado) como no parâmetro "AMP", até de 0 a 32768, como no parâmetro "MErange". Ainda, alguns destes parâmetros como o "CTU" que modifica o tamanho máximo das CTUs, está atrelado aos tamanhos de estruturas aceitos pelo codificador, sendo desta forma discretos e aceitando apenas valores específicos como 16, 32 e 64. Estas características específicas de cada parâmetro acabam tornando difícil a aplicação de certos tipos de *crossover*, como por exemplo o aritmético, que poderia ser aplicado sem problemas no parâmetro "MErange", mas necessitaria de regras especiais para parâmetros discretos como o "CTU" e ainda não teria efeito se aplicado no "AMP". Este comportamento acabaria gerando uma maior influência indesejada de determinados parâmetros nos indivíduos filhos.

Também vale ressaltar que os parâmetros que aceitam apenas duas configurações (liga/desliga) não recebem um inteiro como entrada, mas sim tem sua chamada alterada. Por exemplo, para desativar o parâmetro "AMP" passa-se o parâmetro "No-AMP" para o codificador. Outro aspecto importante do comportamento dos parâmetros é que embora as suas ordens em uma configuração não importem, alguns parâmetros possuem influência muito maior na eficiência de codificação do que outros, o que limita o número de interações e aumenta o impacto dos vieses de distribuição e proporção causados por *crossovers* como o de ponto único.

O primeiro passo para começar a definir o funcionamento do AGMO proposto foi encontrar um método de *crossover* que se adaptasse eficientemente às complexas restrições impostas pelo funcionamento dos parâmetros do codificador x265. Os métodos de *crossover* de ponto único, multi-ponto, uniforme e aritmético discutidos anteriormente foram experimentados sob as mesmas circunstâncias e demais operadores genéticos para assim ser possível identificar qual método possui melhor influência na conversão dos resultados do AGMO. Primeiramente, devido à restrição de tamanho da população causada pela alta demanda temporal de uma codificação, a grande diferença na influência entre os parâmetros e de suas próprias entradas, as técnicas de ponto único e multi-ponto se mostraram incapazes de fazer o AGMO convergir para resultados aceitáveis em tempo hábil. Para tornar tais técnicas mais eficientes neste

contexto seria necessário classificar os parâmetros de acordo com seu grau de impacto na codificação para balancear a influência de cada pai no seu filho. Contudo, estes impactos podem mudar de um codificador para outro e necessitam de um especialista para realizar o balanceamento, o que vai contra a intenção de generalidade do AGMO sendo proposto.

Ademais, apenas o *crossover* heurístico foi aplicado dentre os métodos focados em números inteiros pela sua possibilidade de fazer com que todos os genes do indivíduo filho sejam uma média exata entre os genes de ambos os pais ao se utilizar uma taxa de 0,5. Esta taxa se torna interessante devido ao fato de garantir que o filho não seja influenciado demasiadamente por um dos pais e possibilitar a sua aplicação em parâmetros como o "CTU" cujas entradas são discretas. Porém, mesmo ao se utilizar esta taxa, a técnica se mostrou ineficaz por não levar o algoritmo a convergir para uma solução sub-ótima.

Os demais métodos tornam-se desinteressantes devido à aplicação de números aleatórios distintos para os genes. Pois esta aleatoriedade pode causar com que a diferença entre o impacto dos parâmetros torne na maioria das vezes a influência de um dos pais no filho muito superior do que a do outro. Desta forma estas técnicas não foram avaliadas devido a sua incompatibilidade com o cenário em estudo.

Como esperado devido às suas características de ser propício para populações pequenas e de diminuir os vieses de distribuição e proporção causado pelos diferentes impactos dos parâmetros, o método de *crossover* uniforme mostrou a melhor influência na conversão do algoritmo para soluções ótimas e na distribuição genética da população.

Após o *crossover* ser definido, foi necessário encontrar uma métrica ou técnica que nos permitisse correlacionar as três variáveis desejadas para servir como valor de *Fitness* e nos permitisse na operação de seleção determinar o nível de consumo energético desejado. A primeira métrica avaliada foi a  $WN_{ERD}$  (*Weighted Normalized Energy Compression Distortion*) proposta por MONTEIRO et al. (2014) que correlaciona as informações de energia, compressão e qualidade da imagem em um único valor através da Equação 4. Como pode ser observado, esta métrica atribui um peso a cada uma destas variáveis através da variável  $\alpha$ , sendo que o valor dos pesos de cada variável deve totalizar três. Representar estes três eixos em um único valor livraria o problema atual de sua característica multiobjetivo, contudo, apesar da aplicação desta métrica como valor de *fitness* ter sido experimentada com diversos pesos distintos para a variável de energia, não houve êxito em se controlar o nível de economia de energia que cada execução do algoritmo deveria atingir.

$$WN_{ERD} = (\alpha_{BR} * BR_{normalizado}) +$$

$$(\alpha_{PSNR} * PSNR_{normalizado}) +$$

$$(\alpha_{Energia} * Energia_{normalizado}))$$

$$\text{Onde} \quad \alpha_{BR} + \alpha_{PSNR} + \alpha_{Energia} = 3 \quad (4)$$

Desta forma, em uma segunda tentativa de manter o problema mono-objetivo trocamos o  $WN_{ERD}$  pela métrica de RD (Rate-Distortion). Esta métrica é utilizada internamente no codificador com o objetivo de determinar as decisões de quais estruturas e ferramentas utilizar em cada bloco do sendo codificado e, portanto, mostra-se adequada para avaliar também a eficiência de uma configuração em relação as suas eficiências de compressão e de qualidade de imagem. Ainda em uma tentativa de tornar o problema mono-objetivo, utilizamos os valores de energia como peso aplicado à métrica RD de forma que quanto mais longe do SP desejado a configuração estivesse, pior se tornaria o seu RD ou seja, seu *fitness*. Embora diversos pesos tenham sido aplicados à energia para que esta possuísse influência o suficiente no valor do RD, a técnica resultou em uma grande instabilidade na conversão das simulações para os consumos desejados pois, na maioria das vezes, o algoritmo convergia para configurações com uma redução energética de 98%.

Após as falhas em evitar que a busca se tornasse multiobjetivo, decidiu-se tornar o AG em um AGMO e utilizar o RD em conjunto com os valores de energia e controlar a conversão do consumo das configurações para o SP desejado através das operações de seleção. Nos experimentos realizados avaliando os valores de *fitness* mono-objetivos, foram avaliadas as técnicas de seleção de Roleta, Classificação e Torneio. Ao se utilizar a métrica  $WN_{ERD}$ , o método de roleta se mostrou o mais eficaz devido a baixa variação entre os valores da métrica. Já o RD é representado por valores muito grandes e conseqüentemente apresenta uma grande variação nos valores de *fitness* das configurações, o que levou o método de classificação a obter os melhores resultados.

Ao se ter duas variáveis para levar em consideração, cada configuração pode ser representada por um ponto em um plano 2D que representa o RD e a energia de cada configuração. Tendo este plano pode-se aplicar métodos de seleção como Pareto juntamente com ranqueamentos como os propostos por GOLDBERG (1989) e FONSECA; FLEMING (1998) ou ainda o método de *Niched Pareto* proposto por HORN; NAFPLIOTIS; GOLDBERG (1994). Contudo, nenhuma das técnicas mostrou-se eficaz no controle sobre a convergência do consumo das configurações para o SP desejado.

Desta forma foi desenvolvida uma nova técnica de seleção baseada no método de *Niched Pareto* que nos permitiu controlar com precisão o consumo final das configurações encontradas pelo AGMO. Esta técnica será abordada profundamente na seção seguinte.

Por fim, ainda foram avaliados diversas taxas e métodos de mutação. A primeira mutação aplicada selecionava indivíduos e genes aleatoriamente e foi avaliada com probabilidades de ocorrer que variavam entre 0.5% até 30%. Devido ao fato da permutação entre os parâmetros não possuir impacto na eficiência da codificação e, conseqüentemente, causar uma grande queda na diversidade genética da população, as probabilidades de mutação mais altas se mostraram mais úteis para ajudar o algoritmo a convergir para configurações sub-ótimas. Todavia, para acelerar a conversão das soluções para o SP desejado, uma técnica de mutação mais especializada foi proposta neste trabalho. As técnicas de mutação e seleção desenvolvidas neste trabalho serão explicadas juntamente com a definição do AGMO no Capítulo 4.

O RDE-AGMO funciona através de um *loop* repetido a cada geração onde o AGMO busca melhorar a eficiência das configurações entre as gerações até encontrar uma solução sub-ótima aceitável ou ter sido repetido mil vezes (avaliado mil gerações). Na primeira etapa deste *loop* o algoritmo gera a população inicial de forma aleatória. Desta maneira cada parâmetro de uma configuração pode receber qualquer uma de suas configurações possíveis. Cada população no AGMO é composta por quarenta indivíduos distintos. Este tamanho populacional foi definido através de vários experimentos por prover as melhores convergências em relação ao tempo levado pela execução do algoritmo. Mesmo com este tamanho populacional restringido, o algoritmo avalia cerca de 40.000 configurações caso seja executado até o seu ponto de parada de mil gerações. Número aproximadamente 165 vezes maior do que a quantidade de configurações avaliadas por CORREA et al. (2015), que é o controlador para codificadores HEVC que possui a análise de sensibilidade mais extensa dentre os estudados.

Após gerar a primeira população o algoritmo se encarrega de codificar todas as configurações geradas e coletar dados sobre a distorção nas amostras de luma do vídeo, a sua taxa de *bits* e o seu consumo energético através da codificação de um vídeo. As informações sobre as amostras de luma e a taxa de *bits* são então utilizadas para calcular o RD que será utilizado juntamente ao consumo da codificação como valor de *fitness* da configuração, distribuindo assim os indivíduos em um plano RDE (*Rate-Distortion-Energy*) como o apresentado na Figura 12.

A Equação 5 demonstra o cálculo realizado para se obter os valores de RD. O cálculo desta métrica é composto pela soma das médias dos valores de distorção nas amostras de luma e da taxa de bits de todos quadros codificados, sendo que a segunda variável é multiplicada pelo Multiplicador de Lagrange ( $\lambda$ ). Logo, é possível

afirmar que quanto menor for o resultado deste cálculo, melhor é a eficiência da codificação. Ainda,  $\lambda$  utilizado na equação é um número arbitrário definido pelo codificador x265 que varia de acordo com o *Quantization Parameter* (QP) sendo utilizado.

$$RD = \text{Média}_{Luma} + \lambda * \text{Média}_{TaxadeBits} \quad (5)$$

QP é um parâmetro que define a intensidade com que a quantização remove informações visuais menos perceptíveis ao olho humano, porém que possui grande impacto na qualidade do vídeo sendo codificado. As *Common Test Conditions* (CTCs) (ITU-T, 2015) definem quatro QPs distintos para serem usados com o intuito de avaliação e comparação de resultado, com estes sendo: 22, 27, 32 e 37. Neste trabalho todas as configurações serão buscadas utilizando o QP 22 como padrão por este ser o que causa menor perdas visuais na codificação.

Ao término da codificação de todas 40 configurações da população inicial teremos cada indivíduo disposto como um ponto em um gráfico 2D que relaciona os seus valores de RD e consumo energético normalizado como o apresentado na Figura 12. Como podemos observar no gráfico, devido ao comportamento dos parâmetros do codificador, existe uma tendência das configurações se concentrarem em uma região do plano, o que indica que possuem características parecidas. Para lidar com isso foi desenvolvido um método de seleção de torneio por nicho que torna a avaliação das configurações mais especializada em ambos eixos.

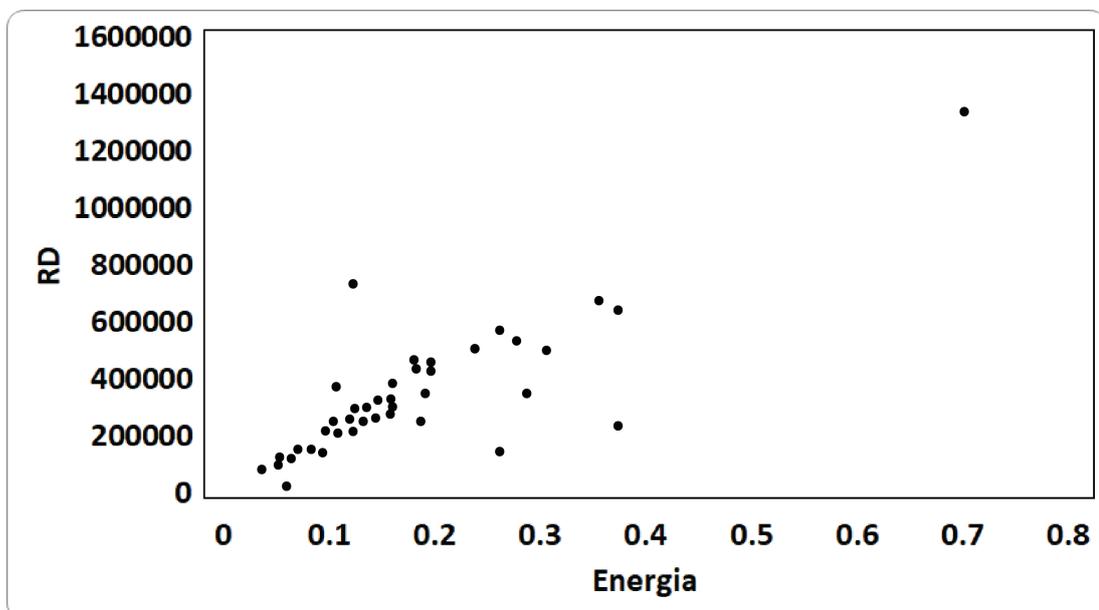


Figura 12 – Distribuição RDE da Primeira População

A operação de seleção é responsável por gerar todos indivíduos de uma próxima geração. Desta forma, esta operação é fundamental para a evolução da população do algoritmo em direção aos objetivos desejados.

## 4.2 Seleção

O método de seleção no AGMO proposto tem de lidar com múltiplos desafios para viabilizar uma convergência aceitável para o algoritmo. Entre estes desafios estão a necessidade de direcionar a convergência das soluções do algoritmo para o consumo esperado enquanto maximiza a eficiência da codificação, dar chances viáveis de seleção a todos os indivíduos devido a grande distância entre seus valores de *fitness*, garantir uma convergência rápida e, também, preservar o melhor resultado.

Primeiramente, para garantir que o melhor resultado encontrado pelo algoritmo seja sempre preservado, foi aplicado o conceito de elitismo. Com isto, a seleção transmite o melhor indivíduo presente na geração sem modificações para a próxima geração. A seleção considera como melhor indivíduo aquele que possui o menor RD e que está dentro de uma faixa de consumo desejado definido pelo usuário. Em seguida, o operador de seleção escolhe através de um processo de classificação seguido de um torneio por nicho os pares que irão se reproduzir para gerar os demais indivíduos da próxima geração.

A Figura 13 apresenta os passos desta etapa pós-elitismo do operador de seleção proposto para o RDE-AGMO e será utilizada como base para a explicação do seu funcionamento. No primeiro passo da seleção (I) o algoritmo sorteia aleatoriamente dois pares de indivíduos na população de acordo com seus valores de energia. Os dois indivíduos do primeiro par estão marcados na cor vermelha. Para garantir que a seleção direcionasse o RDE-AGMO para soluções com o nível de consumo desejado, implementamos um sistema de pesos similar ao proposto por GOLDBERG (1989). Todavia, como o critério de avaliação nesta etapa da seleção é apenas o consumo energético das configurações, pesos maiores foram atribuídos às configurações presentes dentro de uma faixa de consumo desejado ao invés de indivíduos não dominados. A faixa de consumo admite um erro de 5% em relação ao SP desejado, ou seja, se o algoritmo estiver buscando uma configuração que gere uma redução energética de 30%, as configurações que possuírem reduções entre 25% e 35% receberão um peso maior do que as demais. Esta área de consumo contendo os indivíduos com maior peso na seleção está representada na figura pela área demarcada em cinza. Diversos pesos foram avaliados ao longo do desenvolvimento do algoritmo, com os pesos mais eficientes se mostrando o de três para configurações com o consumo desejado e de um para as demais. Desta forma, as configurações dentro da faixa de economia admitida possuem probabilidade três vezes maior de serem selecionadas do que as demais, probabilidade esta que conseqüentemente implica que as futuras configurações tendam a estar nesta faixa de consumo.

Após a seleção de ambos pares de indivíduos de acordo com seus consumos energéticos, um indivíduo de cada par deve ser selecionado para se reproduzir. Esta

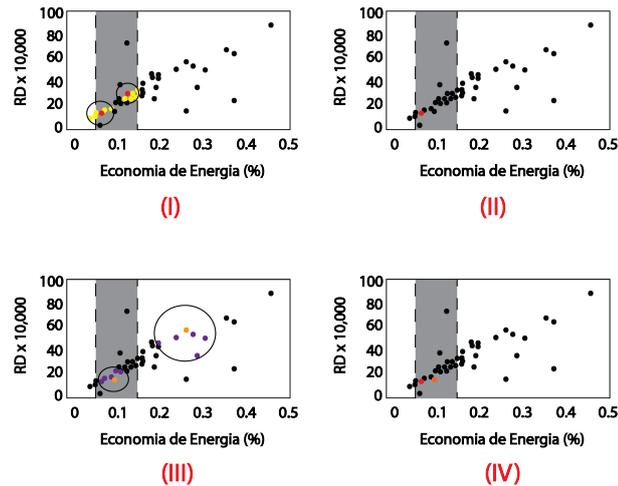


Figura 13 – Passos do Operador de Seleção

seleção por sua vez deve garantir que as configurações com melhores eficiências RD sejam selecionadas, assim como dar probabilidades justas de escolha para ambas. Para atingir ambos objetivos desenvolvemos uma seleção por torneio similar a de HORN; NAFPLIOTIS; GOLDBERG (1994).

A aplicação do conceito de dominância de Pareto é inapropriada nesta etapa de seleção pois relacionaria os valores de RD com o eixo de Energia sendo que o estágio anterior já direcionou as configurações em relação ao seus consumos, podendo assim desviar a busca em relação ao consumo das configurações e não dar prioridade o suficiente para a eficiência RD das configurações. Desta forma, aplicamos uma competição de torneio entre cada par de indivíduos utilizando os valores de RD como base para comparação.

Neste torneio ambas configurações do par tem seu RD comparado com o de cinco outras configurações selecionadas aleatoriamente em uma área próxima a ela. Então é contabilizado quantas destas cinco configurações possuem RD maior que o da configuração competindo. As configurações selecionadas para a comparação dos indivíduos do primeiro par estão demarcadas em amarelo. Note que todas configurações selecionadas estão dentro da área demarcada pelas elipses de cada indivíduo participando do torneio, pois estas demarcam as configurações que podem ser selecionadas para comparação.

Em seguida, a configuração competindo que possuir o maior número de configurações com RD maior que o seu, que neste exemplo esta demarcada de vermelho em (II), é selecionada para reproduzir com a configuração campeã do segundo par. Posteriormente, dois novos indivíduos (demarcados em laranja) são selecionados para passarem pelo mesmo processo (III). No final, tem-se um indivíduo de cada par selecionado (IV) (o indivíduo vermelho do primeiro par e o laranja do segundo), que então trocarão genes em uma reprodução realizada pela operação de *crossover* para assim

gerar um novo indivíduo.

### 4.3 Crossover e Mutação

No *crossover* os genes de ambos indivíduos selecionados para reproduzir são mesclados de forma a gerar um novo indivíduo que possua genes de ambos pais. Como foi visto no capítulo 3, o operador de *crossover* deve lidar com os vieses de distribuição e proporção fortemente presentes nas configurações do codificador devido a diferença nas influências de cada parâmetro. Além disto, esta operação genética sofre a influência de outros fatores como o restringido tamanho populacional adotado.

Frente a estas características, o método de *crossover* que demonstrou maior eficiência foi o *Crossover* uniforme. Desta forma, após a seleção, o RDE-AGMO percorre todos os parâmetros dos indivíduos e seleciona aleatoriamente qual dos pais irá transmitir o parâmetro atual. Existem propostas de *crossover* uniforme que propõem chances de seleção maiores para o pai com maior valor de *fitness*. Contudo, devido ao volátil comportamento dos parâmetros do codificador, utilizar chances iguais para transmitir os genes acaba se tornando mais favorável. A aleatoriedade proporcionada por esta estratégia garante com que de maneira geral ambos pais possuam influência moderada nos filhos. A Figura 14 exemplifica a aplicação do *crossover* uniforme no algoritmo desenvolvido considerando os primeiros cinco genes. Vale ressaltar também, que gerar apenas um filho por *Crossover* resultou em uma melhor distribuição genética dos indivíduos frente as restritas características de seus genes.

Pai 1	CTU 32	AMP 1	MErange 87	No-rect	Early-skip	rdoq-level 2	limit-tu 2
Pai 2	CTU 64	AMP 0	MErange 64	Rect	Early-skip	rdoq-level 1	limit-tu 3
Filho	CTU 32	AMP 0	MErange 87	No-rect	Early-skip	rdoq-level 2	limit-tu 3

Figura 14 – Operador de *Crossover* Uniforme

Todos os genes possíveis avaliados pelo algoritmo que poderiam participar do código genético de um indivíduo estão apresentados na Tabela 7. Note que as tabelas de parâmetros e configurações apresentadas na seção 2.1 apresentam faixas maiores de configurações possíveis para alguns dos parâmetros. Esta diferença ocorre pois neste trabalho decidiu-se utilizar apenas os valores de parâmetros utilizados nas configurações padrões do x265 que também foram utilizados por MACHADO (2017) para assim se ter uma base de comparação mais justa entre as metodologias.

Tabela 7 – Parâmetros avaliados com suas respectivas configurações

Parâmetro	Configurações Avaliadas
CTU	32, 64
Min-CTU-Size 8	8, 16, 32
Max-Merge	2...5
RC-LookAhead	10, 15, 20, 25, 30, 40, 60
RD	2,3,4,6
Rdoq-Level	0,2
Tu-Intra	1...4
TU-Inter	1...4
SceneCut	0,40
Ref	1...5
Limit-refs	3...0
ME-Range	32, 57, 64, 92
Subme	0...5
ME	dia, hex, star
AQ-Mode	On/Off
Rect	On/Off
Amp	On/Off
Limit-Modes	On/Off
Early-Skip	On/Off
Fast-Intra	On/Off
B-Intra	On/Off
SAO	On/Off
SignHide	On/Off
Weight-p	On/Off
Weight-b	On/Off

#### 4.4 Mutaç o

A muta o   respons vel por manter a diversidade gen tica da popula o assim como ajudar o algoritmo a escapar de m nimos locais. Devido ao tempo necess rio para se codificar cada configura o, tamb m desenvolvemos uma heur stica de muta o que acelerasse a velocidade de converg ncia do algoritmo.

Para obter tal influ ncia, a muta o foi aplicada nos indiv duos presentes nas extremidades do consumo energ tico da popula o se esta estiver completamente acima ou abaixo da faixa de consumo desejada. Desta forma, caso ocorra muta o e todos os indiv duos estejam consumindo menos do que a faixa desejada, os dois indiv duos com os menores consumos ter o um de seus par metros alterados de forma que os seus consumos aumentem. O inverso ocorre se todos os indiv duos tiverem um consumo superior ao estabelecido pela faixa de consumos aceit veis estabelecida. Por fim, se houverem indiv duos dentro da faixa definida, a muta o ocorre de forma totalmente aleat ria em um indiv duo, ou seja, apenas um indiv duo   selecionado

Tabela 8 – Diferentes avaliações para cada aspecto do algoritmo

<b>Operação</b>	<b>Diferentes Avaliações</b>
Tamanho da População	<b>40</b> , 80, 120, 160
Métodos de Seleção	Roleta, Torneio, <b>Classificação por Nicho</b>
Métodos de Crossover	<b>Uniforme</b> , 1-Ponto, 2-Pontos, Flat
Mutação	<b>Simples, Direcionada</b>
Taxas de Mutação	0,5%, 1%, 2%, <b>3%</b> 5%, 10%
Peso de Classificação na Seleção	1, <b>3</b> , 5, 10
Métricas de Fitness	WNerd, <b>RD / Energia</b>

aleatoriamente para sofrer mutação e um dos seus parâmetros é alterado de forma também totalmente aleatória, independentemente se isso irá aumentar ou diminuir o seu consumo.

De acordo com OCHOA; HARVEY; BUXTON (2000), as probabilidades ótimas de que ocorra a mutação tendem a aumentar juntamente com a pressão de seleção. Visto que o tamanho populacional utilizado não é extenso, e que medidas foram tomadas para acelerar a convergência do algoritmo, a pressão de seleção do mesmo acabou se tornando elevada. Como consequência, a probabilidade de mutação que fez o algoritmo convergir mais frequentemente para soluções eficientes foi de 3%.

Na Tabela 8 podem ser observados todos métodos e parâmetros avaliados neste trabalho, com aqueles em negrito sendo os selecionados para uso no estado final do algoritmo desenvolvido.

Uma vez que todos os operadores genéticos do algoritmo foram definidos e implementados, o mesmo foi utilizado para encontrar configurações energeticamente eficientes para servirem como PCs de um controlador. Em seguida, o comportamento do algoritmo e o impacto da inserção das configurações encontradas em um controlador de energia foram estudados e discutidas no capítulo 5.

## 5 RESULTADOS

Para avaliarmos a eficiência do algoritmo desenvolvido é de extrema importância analisarmos o seu comportamento durante a execução e o impacto dos PCs encontrados. Para isso, o RDE-AGMO foi executado com a finalidade de encontrar configurações de alta eficiência e determinados consumos energéticos em um computador com um processador Haswell i7-4790k 4Ghz, 16GB de RAM operando sob o sistema operacional Ubuntu 16.04 LTS. Para averiguarmos a eficiência destas configurações será utilizado a métrica *BD-Bitrate* (BD-BR). Esta métrica baseada na *BjØntegaard Difference* (BD) (BJØNTEGAARD, 2008) determina a degradação na eficiência de codificação entre duas configurações através da correlação das suas a distorções (PSNR) e taxas de *bits* (*BitRate*). Desta forma, um BD-BR de 2,5% indica que a configuração teve um aumento de 2,5% na taxa de bits considerando uma qualidade de vídeo (PSNR) constante.

Como mencionado anteriormente, as características de um vídeo possuem grande impacto na eficiência dos parâmetros na codificação do vídeo. Uma das características com maior impacto é a sua resolução e, em vista disso, as CTCs classificam os vídeos de teste em classes de acordo com as suas resoluções. Desta forma, se utilizarmos um vídeo HD como base para obtermos as configurações, o algoritmo irá encontrar soluções sub-ótimas para vídeos HD, mas possivelmente não tão eficientes para vídeos com menores resoluções. Logo, este trabalho buscou dois conjuntos de PCs denominados de *Configuration Sets* (CSs), o primeiro utilizando um vídeo com resolução 832x448, e outro utilizando um vídeo com resolução 1920x1080p. Para tornar a avaliação justa, cada conjunto terá sua avaliação realizada com vídeos das CTCs pertencentes na mesma classe, ou seja, que possuam resoluções similares.

Primeiramente o algoritmo foi utilizado para encontrar nove configurações distintas para vídeos de resolução 832x448 utilizando o vídeo *PartyScene* indicado pelas CTCs como classe C. Em seguida, para destacar o impacto de diferentes distribuições de PCs no controle, um conjunto de quatro PCs foram encontrados utilizando o vídeo *Kimono* de resolução 1920x1080 pertencente a classe B. Estas configurações então podem ser inseridas como PCs no controlador desenvolvido por MACHADO (2017)

para observarmos o impacto no comportamento do controle causado por esta inserção. Com tal observação é possível determinarmos a substituição da metodologia de análise de sensibilidade anteriormente utilizada por MACHADO (2017) pelo algoritmo sendo proposto. Para tornar o estudo sobre este impacto mais justo, o codificador foi utilizado sob a configuração *Low Delay* empregada no desenvolvimento do controle de MACHADO (2017).

Por fim, uma das características mais importantes que o AGMO proposto deve proporcionar às suas soluções e que outras metodologias carecem é a possibilidade de selecionar com precisão o nível de economia que os PCs devem ter em relação à configuração base (x265 *Preset 8*). Embora o número e a distribuição de PCs possam mudar de controle para controle, considerou-se inicialmente neste trabalho o cenário de um controlador que possua 9 pontos de controle com economias distribuídas uniformemente cobrindo uma faixa de economias de 10% a 90%. Desta forma, nove configurações foram geradas com distâncias de 10% entre suas economias. Cada configuração foi nomeada de acordo com a porcentagem de economia pretendida, ou seja, a que proporciona 10% de economia foi denominada PC10 e a que providencia 90% de PC90.

Desta maneira, em 5.1 serão apresentados os resultados do comportamento do algoritmo ao longo de suas gerações em busca de diferentes PCs. Em seguida, na seção 5.2 os PCs encontrados pelo algoritmo serão avaliados e comparados com os PCs encontrados pelas demais metodologias. Por fim, em 5.3 os resultados sobre o impacto da inserção dos PCs encontrados no controlador de energia desenvolvido por MACHADO (2017) que servirá como *baseline* de comparação.

## 5.1 Comportamento e Convergência do RDE-AGMO

Um dos principais comportamentos a serem examinados em um algoritmo genético é a sua convergência, pois esta mostra o avanço realizado ao longo das gerações em direção ao resultado desejado. Para observar este comportamento podemos observar o desvio padrão das médias dos consumos energéticos e RDs dos indivíduos de cada geração. Dado que ao evidenciar a homogeneidade da população de cada geração, nos permite avaliar o quão próximo de chegar a um resultado o algoritmo está.

A Figura 15 apresenta o desvio padrão ao longo das gerações criadas na execução do algoritmo em busca da configuração PC30 para o vídeo *PartyScene*. O gráfico nos mostra que as duas primeiras gerações possuem os maiores desvios padrão na execução do RDE-AGMO, apresentando um desvio de aproximadamente 270 mil de RD na segunda geração. Este alto desvio inicial ocorre devido à primeira população ser completamente aleatória. Já nas próximas gerações o desvio padrão decai e se mantém similar até a última geração, evidenciando a alta pressão de seleção do algoritmo.

Também é possível ver que em certas gerações como a 25, 37 e 45, ocorrem leves aumentos no desvio padrão. Estes picos são causados naturalmente pela ocorrência de mutações e, portanto, são desejados para que o algoritmo escape de mínimos locais.

Outro aspecto importante do algoritmo é a sua capacidade de convergir para soluções que possuam os níveis de consumo energético dentro do desejado. As Figuras de 16 e 17 apresentam a evolução das gerações em relação aos seus consumos para as execuções do algoritmo buscando SPs que providenciem 50% e 90% de economia de energia, respectivamente. Nestas figuras a linha laranja representa o consumo ideal em joules a ser atingido (*SetPoint*), as linhas verdes demarcam limites superiores e inferiores de consumos aceitáveis assim como definidos na etapa de seleção, ou seja, representam um erro tolerável de 5% em relação ao SP. Já a linha azul apresenta a média do consumo energético dos 40 indivíduos presentes em cada geração. As demais figuras apresentando os comportamentos dos outros PCs definidos se encontram no Apêndice A.

Analisando o comportamento do algoritmo para todos os PC podemos afirmar que o algoritmo convergiu para soluções dentro da faixa de consumo desejada nas execuções de todos nove *SetPoints* com exceção apenas do SP40, cuja última população se encontra levemente abaixo do limite inferior de consumo. Vale ressaltar que apesar da média populacional estar abaixo da faixa estabelecida, a configuração final selecionada para o SP40 ainda esteve próxima o suficiente do consumo desejado, como será exibido a seguir. Este comportamento estável do RDE-AGMO valida a eficiência do sistema de pesos atribuídos na primeira etapa da seleção juntamente com a capacidade do algoritmo de encontrar configurações com qualquer nível de economia.

As Figuras 16 e 17 mostram também que devido a sua criação aleatória, a população inicial tende a possuir um consumo entre 300 e 500 joules, o que esta consideravelmente à baixo da maioria dos SP desejados. Por exemplo, o consumo da primeira

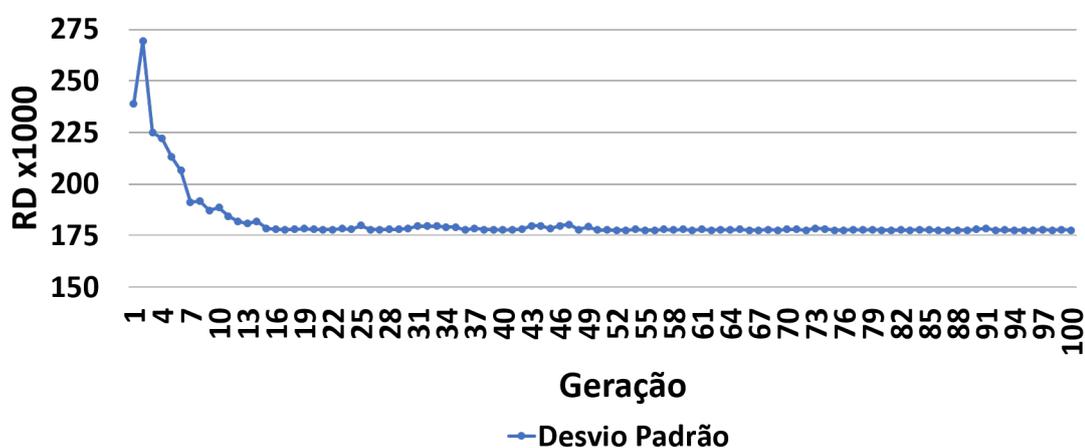


Figura 15 – Desvio Padrão das Gerações Buscando o PC30 para o Vídeo PartyScene

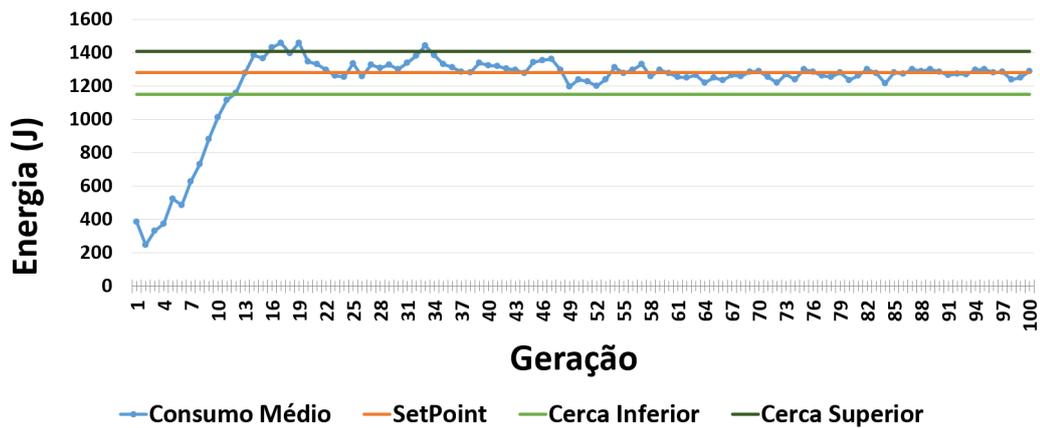


Figura 16 – Média de Consumo das Gerações Buscando o PC50 para o Vídeo PartyScene

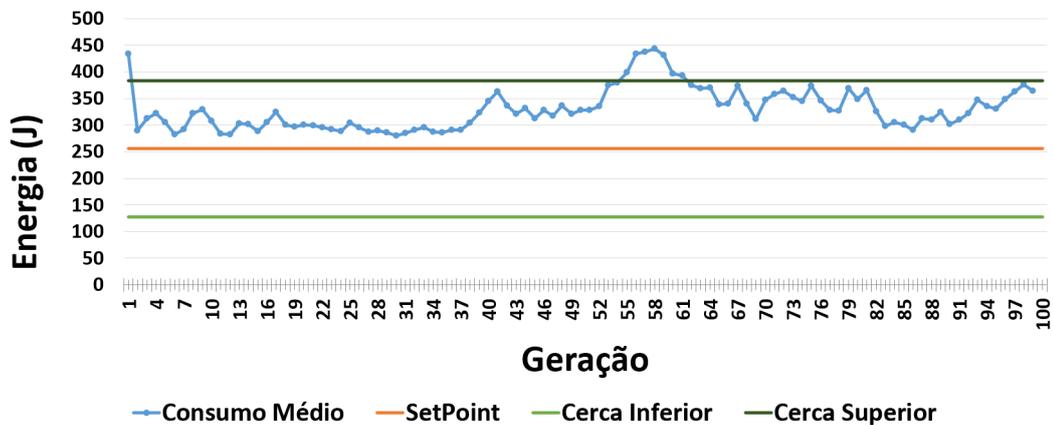


Figura 17 – Média de Consumo das Gerações Buscando o PC90 para o Vídeo PartyScene

geração da busca pelo SP50 está aproximadamente 1000 joules abaixo do SP. Já no SP90 este consumo se torna superior ao *SetPoint*, cujo limite superior é de 380 joules, 50 joules a menos que o consumo médio da primeira população. Este comportamento é repetido na busca de todos os PCs, com o consumo da primeira geração se aproximando gradualmente do SP conforme o nível de economia do PC aumenta. Contudo, devido à criação aleatória da população inicial, nada impede que este comportamento se altere em outras execuções.

Ademais, apesar da distância no consumo médio das populações iniciais, a evolução das gerações em direção ao SP desejado é evidente. Também fica claro que esta distância inicial do consumo implica que quanto maior o consumo do SP desejado, mais gerações são necessárias para que a população esteja dentro da faixa de consumo admitida. Frente a isto, a população da execução buscando o SP10, que possui a menor economia, atinge o consumo desejado apenas na 19ª geração, enquanto o SP20 atinge na 12ª geração, o SP50 na 9ª até o SP90 que atinge na 2ª geração.



Figura 18 – Desvio Padrão das Gerações Buscando o PC20 para o Vídeo Kimono

Avaliando as Figuras 16 e 17 mais detalhadamente, podemos notar que enquanto o consumo médio do PC50 se mantém muito próximo ao SP ideal durante grande parte das gerações, a do PC90 se distancia do objetivo na 54<sup>a</sup> geração, se reaproximando apenas na 62<sup>a</sup> geração. Reafirmando a atuação das mutações na prevenção de mínimos locais. Enfim, apesar da alta taxa de mutação por indivíduo, todas as execuções apresentaram comportamentos estáveis nos consumos de suas gerações.

Por conseguinte, o RDE-AGMO foi utilizado sem modificações para encontrar PCs especializados em vídeos com resolução *Full HD* (1920x1080p) empregando o vídeo *Kimono* como base para codificação e cálculo de *Fitness*. Desta forma, também cabe analisarmos o comportamento do RDE-AGMO quando aplicado a vídeos com resoluções maiores.

A Figura 18 ao apresentar o desvio padrão do RD das gerações na busca pelo PC20 expõe que, quando colocado a encontrar configurações especializadas em vídeos Full HD, o RDE-AGMO apresenta gerações menos homogêneas. Ademais, o gráfico também mostra que a conversão do algoritmo foi menos linear neste caso, com os desvios padrão apresentando diversos picos ao longo da evolução das gerações. Todavia, o RDE-AGMO continuou convergindo em todos os casos para RDs menores, mesmo com a leve alteração na homogeneidade das gerações.

As Figuras 19 e 20 apresentam a média do consumo energético de cada geração do RDE-AGMO ao buscar os PCs 20 e 60 para o vídeo *Kimono*. Pelo fato do vídeo *Kimono* ter uma resolução muito maior que o *PartyScene*, o RDE-AGMO acabou necessitando de um maior número de gerações para atingir o consumo desejado. Conforme pode ser visto na Figura 19, a população do RDE-AGMO só atingiu o consumo aceitável na 40<sup>a</sup> geração, sendo que esta meta foi atingida na 13<sup>a</sup> geração para o mesmo PC do vídeo *PartyScene*. Este tempo para atingir o consumo desejado demonstra que um aumento no número máximo de gerações pode gerar maiores ganhos quando o RDE-AGMO for aplicado a vídeos com resoluções maiores, como

por exemplo, 4k. Analisando os gráficos podemos notar também que o RDE-AGMO converge para os múltiplos *SetPoints* com comportamentos similares aos anteriores mesmo quando aplicado sem modificações ao vídeo Kimono, o que salienta a generalidade de sua utilização.

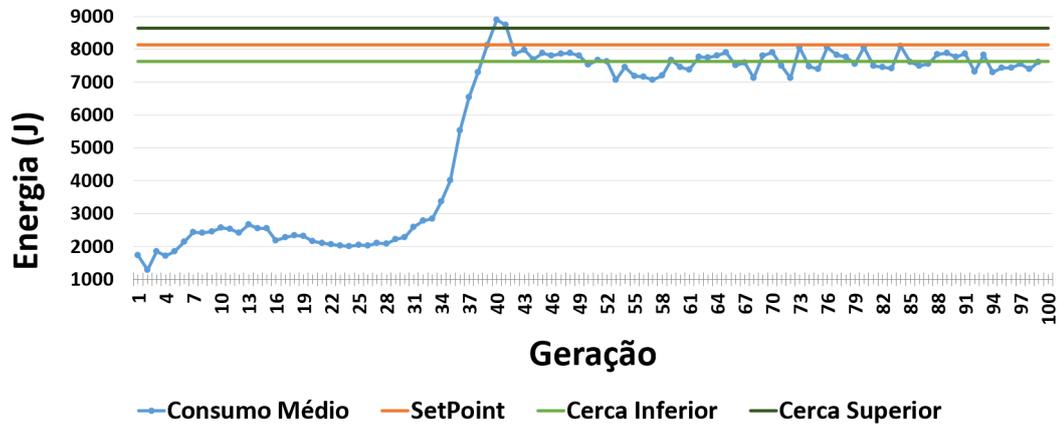


Figura 19 – Consumo Médio das Gerações Buscando o PC20 para o Vídeo Kimono

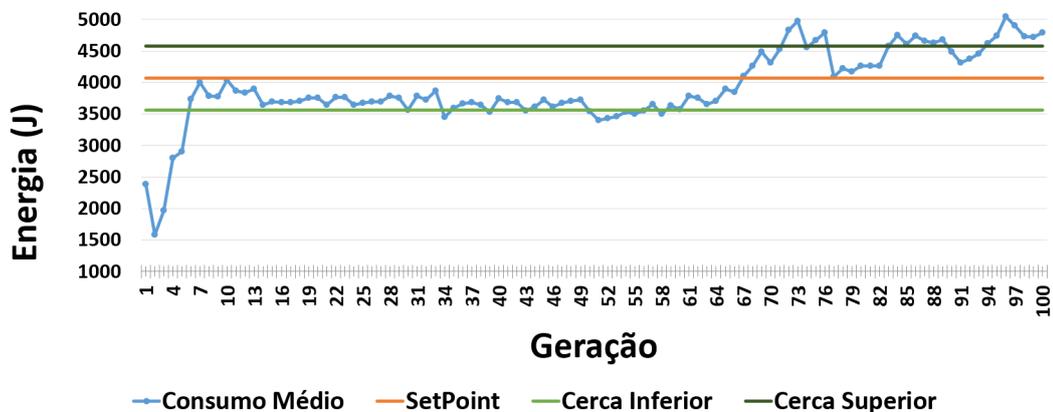


Figura 20 – Consumo Médio das Gerações Buscando o PC60 para o Vídeo Kimono

## 5.2 Eficiências dos Pontos de Controle

Para termos uma melhor análise sobre a eficiência do algoritmo, além do seu comportamento em busca das soluções, também é necessário avaliarmos a eficiência das configurações encontradas. A configuração selecionada pelo algoritmo é aquela considerada como melhor indivíduo ao término da sua última geração. Adicionalmente, a configuração selecionada será utilizada como um Ponto de Controle (PC) que, como definido anteriormente, é uma configuração que visa atingir uma determinada redução energética em um controlador. Para tornar as comparações justas, avaliaremos

Tabela 9 – Parâmetros dos PCs Encontrados para vídeos 448p

	PC10	PC20	PC30	PC40	PC50	PC60	PC70	PC80	PC90
<b>CTU</b>	32	64	32	32	32	32	32	32	32
<b>Min-CU-size</b>	8	8	8	8	8	8	8	8	8
<b>Max-Merge</b>	4	2	4	4	4	2	3	5	3
<b>Lookahead</b>	60	25	10	40	40	20	15	20	25
<b>RD</b>	6	6	6	6	6	6	4	6	4
<b>Rdoq-level</b>	0	0	0	0	0	0	0	0	0
<b>TU-Intra</b>	3	4	4	4	3	2	1	4	2
<b>TU-inter</b>	1	1	3	3	1	1	1	1	1
<b>SceneCut</b>	40	0	0	40	0	27	0	40	0
<b>Ref</b>	5	5	5	5	5	5	5	5	5
<b>Limit-refs</b>	0	0	0	1	0	1	0	2	2
<b>MErange</b>	92	92	32	64	92	32	64	32	32
<b>SubME</b>	4	4	5	2	5	5	4	1	4
<b>ME</b>	star	star	star	star	hex	hex	hex	hex	hex
<b>AQ-mode</b>	0	1	1	0	0	1	0	0	1
<b>rect</b>	1	1	1	1	1	1	1	1	1
<b>AMP</b>	1	0	0	1	0	0	1	1	1
<b>Limit-modes</b>	0	0	0	0	0	0	0	0	0
<b>Early-skip</b>	1	1	1	1	1	1	1	1	1
<b>Fast-intra</b>	0	0	1	0	0	0	0	1	0
<b>B-intra</b>	0	1	1	0	0	0	0	0	1
<b>SAO</b>	0	1	0	0	0	0	0	0	0
<b>Signhide</b>	0	1	0	1	0	1	0	0	0
<b>Weight-p</b>	0	0	1	0	0	0	0	0	0
<b>Weight-b</b>	0	1	1	0	0	0	1	1	1

os PCs encontrados para o vídeo *PartyScene* e *Kimono* com base em vídeos das suas respectivas classes nas CTCs.

A Tabela 9 mostra todos os parâmetros que compõem cada uma das 9 configurações encontradas pelo RDE-AGMO. Ao relacionar múltiplos parâmetros e configurações, é possível concluir que o RDE-AGMO foi capaz de descobrir características sobre as eficiências dos parâmetros para diferentes consumos e características do vídeo. Se analisarmos o parâmetro CTU é possível notar que o algoritmo identificou que um tamanho máximo de CTU de 32 pixels tende a ser mais eficiente para vídeos com resolução 832x448p. Esta maior eficiência é esperada em vista que tamanhos de CTU de 64 pixels são mais adequadas para vídeos com resoluções maiores devido à densidade dos pixels. Já um tamanho máximo de CTU de 16 pixels acaba causando grandes perdas ao ser aplicado em grandes áreas homogêneas de um vídeo 448p.

Por conseguinte, se observarmos o parâmetro ME que é responsável por determinar qual algoritmo de busca será utilizado na estimação de movimento, fica evidente que a busca em estrela (star), por ser mais custosa, se mostrou mais eficiente para

PCs com menores níveis de economia, enquanto o hexagonal (hex) acabou proporcionando melhor eficiência para os PCs mais econômicos. Além disso, o parâmetro *MErange* que determina o tamanho da área de busca da ME, também mostrou uma tendência de ter tamanhos menores para os PCs que atingem maior economia. Comportamento também justificável, em vista do grande custo envolvido na etapa de estimação de movimento e da alta influência do tamanho da janela de busca neste custo.

Por fim, também podemos afirmar que o algoritmo foi capaz de encontrar relações benéficas e não triviais entre parâmetros inter-relacionados. Como por exemplo, segundo a documentação do MULTICOREWARE (2016), entradas maiores do parâmetro *LookAhead* tendem a melhorar a eficiência do parâmetro *SceneCut*. Comportamento identificado e explorado pelo algoritmo, pois podemos observar que há uma propensão do parâmetro *SceneCut* receber entradas maiores juntamente com o *LookAhead*. Desta forma, os PCs cujo *LookAhead* estava à baixo de 40 possuem também um *SceneCut* baixo de 27 ou de 0, com exceção apenas da configuração PC8.

Para explicitar a eficiência do RDE-AGMO em definir o nível de economia de cada PC, a Tabela 10 traz o nível de redução energética atingido por cada um dos 9 PCs encontrados pelas execuções do algoritmo, juntamente com as economias dos PCs definidos por PENNY (2017) e MACHADO (2017). Ainda, a última linha da tabela nos apresenta o Erro Quadrático Médio (*Mean Squared Error* - MSE) dos três trabalhos para salientar o quão próximo estão os PCs do SetPoint desejado.

Primeiramente podemos analisar que as configurações encontradas pelo RDE-AGMO estão muito próximas do consumo ideal, com a maior distância sendo de 3% e pertencendo ao PC60. Já se avaliarmos os PCs determinados pela metodologia de PENNY (2017), podemos notar que estes cobrem uma faixa de economias menor do que o proposto por este trabalho mesmo ao empregar um ponto de controle a mais. Ainda, ao utilizar metodologia similar, MACHADO (2017) define apenas oito pontos de controle que reduzem o consumo energético entre 44% e 68%. Desta forma, podemos constatar que devido a esta baixa faixa de economias abrangidas, o controlador com estes PCs provavelmente não apresentará bom comportamento para *SetPoints* abaixo 30% ou acima de 68%.

Devido a inabilidade das demais metodologias de selecionar o número de parâmetros e suas distribuições de economia, as soluções de PENNY (2017) e MACHADO (2017) apresentam quantias distintas de PCs e cobrem diferentes faixas de economias. Desta forma, para tornar a comparação entre as metodologias o mais justa possível, os erros foram calculados perante o consumo ideal considerando a faixa de economias abrangida e o número de PCs empregados por cada trabalho. Por exemplo, a distância entre as reduções dos PCs de MACHADO (2017) é de 3,25% por serem oito PCs cobrindo uma faixa de economias de 30% (de 40% a 70%).

Tabela 10 – Comparação Entre a Distribuição da Economia dos PCs Propostos pelo RDE-AGMO e dos Demais Controladores

PC	Economia de Energia (%)		
	RDE-AGMO	PENNY (2017)	MACHADO (2017)
PC10	10	7	44
PC20	18	15	46
PC30	29	28	47
PC40	40	34	54
PC50	51	39	56
PC60	57	49	60
PC70	69	59	63
PC80	80	62	68
PC90	90	70	X
PC100	X	79	X
<b>MSE</b>	16	36	41

Analisando os dados da tabela, podemos observar que os PCs encontrados pelo RDE-AGMO apresentam um MSE de apenas 16. Este baixo erro médio quadrático comprova que o algoritmo proposto possibilita que o usuário defina com extrema precisão o nível de economia de cada PC do seu controlador. Em vista das características mencionadas das metodologias atualmente existentes, os demais trabalhos acabam apresentando PCs com distribuições mais dispersas. Esta característica é salientada pelos MSEs de 36 e 41 que, mesmo ao considerarmos faixas menores de economia, são respectivamente 125% e 150% maiores do que o erro atingido pelos PCs encontrados pelo RDE-AGMO. Ainda, é importante ressaltar que se considerarmos que em um cenário ideal os demais trabalhos deveriam cobrir economias de 10% a 90%, o MSE de PENNY (2017) aumentaria para 437 enquanto o de MACHADO (2017) subiria para 2771.

A Tabela 11 apresenta as perdas de eficiência de codificação em BD-BR causada pelas nove configurações encontradas pelo algoritmo utilizando o vídeo *PartyScene* quando aplicadas a todos os oito vídeos de Classe C das CTCs. A Tabela 11 aponta que as configurações de maneira geral apresentam maior eficiência para o vídeo *PartyScene* possuindo um BD-BR médio de 6,38% de perda. Observe também que o PC10 providenciou uma melhora de 2,38% na eficiência de codificação para o vídeo *BasketballDrill* em relação ao *preset 8* definido pelo próprio x265. Contudo, este comportamento resultou em um aumento de 25% no consumo energético da codificação deste vídeo. Em vista de que o vídeo *BasketballDrive* possui a maior movimentação entre os três casos de teste, este comportamento indica que a configuração possui um conjunto de parâmetros que favorece consideravelmente a etapa de estimação de movimento, causando com que esta seja mais extensiva e consequentemente aumentando a eficiência da codificação juntamente com o seu consumo. Entre

Tabela 11 – BD-BRs e Consumos dos Pontos de Controle Encontrados para CTCs 448p

PCs	BasketballDrill		BQMall		PartyScene	
	BD-BR (%)	Consumo (J)	BD-BR (%)	Consumo (J)	BD-BR (%)	Consumo (J)
PC10	-2,38	1666,19	0,90	953,30	0,31	1761,46
PC20	4,69	1090,78	6,14	866,61	3,49	1522,09
PC30	4,47	721,29	6,17	495,25	3,49	1320,41
PC40	8,29	937,21	10,02	663,51	6,76	1209,71
PC50	5,08	579,98	7,01	492,26	2,32	872,03
PC60	9,39	562,26	10,79	416,36	7,4	834,20
PC70	9,04	445,35	11,05	304,42	8,25	597,08
PC80	9,88	246,03	11,65	194,06	8,51	371,06
PC90	18,17	106,48	20,93	91,69	16,89	178,52
<b>Média</b>	7,40	706,18	9,41	497,49	6,38	962,95

os principais parâmetros responsáveis por este comportamento estão possivelmente a área de busca configurada para 92, a ME realizada em estrela e a AMP ligada, pois são as configurações mais altas de três parâmetros que possuem grande influência na estimação de movimento.

Por fim, embora exista uma variação entre os BD-BRs das configurações para diferentes vídeos devido às suas características, a distância desta variação é similar para todos os casos, com a maior diferença sendo de 4,7% para o PC50 entre os vídeos *PartyScene* e *BQMall*. Tal diferença ocorre por esta configuração ter obtido um impacto muito baixo para o vídeo base, sendo 1,17% mais eficiente do que o SP30 cujo consumo energético é 20% superior. Este comportamento mostra que a configuração encontrada consegue ser extremamente eficiente para o vídeo *PartyScene* enquanto mantém perdas similares a de outros PCs para os demais vídeos. Esta baixa variação, juntamente com a distância média máxima de 3,03% entre as os BD-BRs, eliminam a necessidade de se codificar múltiplos vídeos como função de avaliação, pois comprova que as configurações encontradas com base apenas no vídeo *PartyScene* podem ser aplicadas a outros vídeos sem grandes perdas.

Após todos os PCs especializados em vídeos de resolução 448p serem encontrados, cabe avaliarmos também as eficiências dos PCs encontrados para vídeos *Full HD* (1920x1080p). Como estes vídeos demandam um grande tempo para serem codificados devido suas altas resoluções, foram buscados apenas 4 PCs com distâncias de 20% nos seus consumos para tornar viável o tempo de simulação do algoritmo. A Tabela 12 nos mostra que, assim como o esperado, o tamanho de CTU de 64 se mostrou mais eficiente para vídeos com esta maior resolução. Além disso, é possível concluir também que a relação entre os parâmetros *LookAhead* e *SceneCut* não se mantiveram. Enfim, note também que a busca em estrela se mostrou a mais eficaz entre os demais métodos, sendo este o segundo método mais custoso depois do *FullSearch*.

Tabela 12 – Parâmetros dos PCs selecionados pelo RDE-AGMO para o Vídeo Kimono

Parâmetros	Pontos de Controle			
	PC20	PC40	PC60	PC80
CTU	64	64	64	64
Min-CU-size	8	8	8	8
Max-Merge	5	4	5	4
Lookahead	30	40	60	20
RD	6	6	3	4
Rdoq-level	2	2	2	2
TU-Intra	2	4	3	2
TU-inter	4	1	1	1
SceneCut	0	29	1	0
Ref	3	4	4	3
Limit-refs	1	1	1	0
MErange	57	32	64	64
SubME	4	4	4	3
ME	star	dia	dia	star
AQ-mode	1	1	0	1
Rect	1	1	1	0
AMP	0	0	1	0
Limit-modes	1	0	0	0
Early-skip	1	1	1	1
Fast-intra	1	1	0	1
B-intra	0	0	0	1
SAO	1	1	0	1
Signhide	1	1	0	1
Weight-p	0	0	0	0
Weight-b	0	0	0	0

Tabela 13 – Comparação Entre a Distribuição da Economia dos PCs Propostos pelo RDE-AGMO e dos Controladores

PCs	Economia de Energia (%)		
	RDE-AGMO	PENNY (2017)	MACHADO (2017)
PC10	X	7	44
PC20	20	15	46
PC30	X	28	47
PC40	41	34	54
PC50	X	39	56
PC60	60	49	60
PC70	X	59	63
PC80	79	62	68
PC90	X	70	X
PC100	X	79	X
<b>MSE</b>	2	36	41

Ao analisarmos a economia dos quatro PCs encontrados e apresentadas na Tabela 13 podemos notar que apenas duas destas não possuem exatamente o consumo desejado. Além disto, as demais configurações apresentaram erros de apenas 1%, causando um MSE de apenas 2%.

Por fim, Tabela 14 contém as informações sobre os BD-BRs de cada PC encontrado pelo RDE-AGMO para o vídeo *Kimono* quando aplicados à outros vídeos de mesma classe. Comparando os BD-BRs destes PCs com os encontrados com base no vídeo *PartyScene*, podemos constatar que a perda de eficiência causada pelas configurações se tornou consideravelmente menor. Obtendo inclusive ganhos de eficiência de até 7,47% para o vídeo *BasketballDrive* enquanto reduz 40% do consumo energético. É possível concluir também que devido a alta eficiência das configurações para o vídeo *BasketballDrive*, a diferença de BD-BR entre dois vídeos aumentou para 18,13% no PC80 dos vídeos BQTerrace e *BasketballDrive*. Este acréscimo na diferença das eficiências obtidas por um mesmo PC indica uma necessidade de se utilizar múltiplos vídeos como base de *fitness* ao se buscar por configurações para vídeos com resoluções maiores. Outro ponto a ser analisado nestes dados é que apenas o PC60 teve perdas de eficiência significativas superior a 5% para os três vídeos. Este ponto discrepante ocorreu provavelmente por existir mínimos locais difíceis de escapar para atingir resultados bons próximos a 60% de economia. Ainda, este grande aumento no BD-BR pode ter sido causado principalmente pelos parâmetros *SAO* e *SignHide* estarem desativados, em vista que isto não ocorre nos demais PCs.

Como foi discutido nesta seção, o ideal é encontrar PCs o mais próximo dos *Set-Points* quanto for possível. No entanto, o mais importante em um conjunto de PCs de um controlador é a relação entre eles e que haja uma boa heterogeneidade entre cada PC. Essa heterogeneidade é essencial pois cabe ao controlador navegar entre os PCs

Tabela 14 – BD-BRs das Configurações Encontradas para o Vídeo Kimono

	<b>Kimono</b>	<b>BQTerrace</b>	<b>BasketballDrive</b>
<b>PC20</b>	0.23	4,51	-6,87
<b>PC40</b>	0.01	2,46	-7,22
<b>PC60</b>	15.28	19,82	5,86
<b>PC80</b>	2.89	13,34	-4,79

em busca do *SetPoint*. Desta maneira, os impactos de se aplicar os PCs encontrados pelo algoritmo proposto serão avaliados a seguir.

### 5.3 Impacto das Configurações no Controlador de Energia

O RDE-AGMO foi desenvolvido para encontrar configurações eficientes para codificadores de vídeo, podendo assim automatizar e aperfeiçoar o processo de análise de sensibilidade presente da criação de controladores. Desta forma, é essencial avaliarmos o impacto da aplicação das configurações encontradas pelo algoritmo em um controlador para assim validarmos a sua aplicabilidade.

Com este objetivo, os PCs utilizados no controle desenvolvido por MACHADO (2017) foram substituídos pelas configurações encontradas neste trabalho pelo RDE-AGMO sem modificação alguma ao resto do controlador. Por conseguinte, os vídeos das classes B e C presentes nas CTCs foram codificados sob a atuação do controlador do nosso *baseline* (MACHADO, 2017) e com o desenvolvido utilizando o CS (*Configuration Set*) proposto por este trabalho. Tal experimento nos possibilita realizar uma análise precisa sobre o impacto da utilização do RDE-AGMO no desenvolvimento de um controlador.

A Tabela 15 apresenta a Raiz do Erro Médio Quadrático (Root Mean Squared Error - RMSE), ou seja, um descritor referente à distância de cada atuação do controlador em relação ao consumo desejado para os SPs de 30% e 70%, utilizando respectivamente as configurações originais de MACHADO (2017) e as encontradas pelo RDE-AGMO. Se observarmos o RMSE no consumo dos PCs para o SP30 podemos notar que o controle apresenta um aumento de módulo 2 ao se aplicar as configurações encontradas pelo RDE-AGMO. Ao analisarmos os RMSEs consequentes do CS deste trabalho podemos notar que em relação aos PCs originais, os PCs encontrados para a classe C causam maiores perdas de RMSEs nos QPs mais baixos. Já os PCs utilizados na classe B causam maiores perdas nos QPs mais altos. Podemos notar também que o vídeo *BQTerrace* providenciou a maior dificuldade para os PCs providos pelo RDE-AGMO controlarem a energia, com estes possuindo RMSEs maiores para todos os QPs, com exceção apenas do QP 22 cujo o CS de *baseline* gerou um RMSE de 556. Devido a este alto RMSE discrepante, o vídeo *BQTerrace* foi simulado múltiplas

Tabela 15 – RMSEs Causados por Ambos CSs para Todos Casos de Teste

Classe	Vídeo	QP	RMSE			
			SP30		SP70	
			(MACHADO, 2017)	RDE-AGMO	(MACHADO, 2017)	RDE-AGMO
C	BasketballDrill	22	37	48	178	43
		27	32	39	156	37
		32	26	31	135	31
		37	21	20	115	27
	BQMall	22	62	66	201	52
		27	48	32	162	43
		32	37	37	137	32
		37	26	40	116	24
	PartyScene	22	46	98	227	63
		27	43	63	199	57
		32	35	33	165	45
		37	31	28	128	35
B	BasketballDrive	22	515	443	1418	359
		27	327	318	972	260
		32	261	297	791	219
		37	202	266	667	188
	BQTerrace	22	556	262	1391	446
		27	172	274	692	262
		32	143	229	462	163
		37	72	210	358	128
	Cactus	22	331	279	1045	343
		27	248	219	659	182
		32	218	191	565	147
		37	186	198	492	125
<b>RMSE Médio</b>			153	155	476	138

vezes para o QP 22 para garantir que o resultado estava correto.

Já o RMSE de ambos conjuntos de configurações para o SP70 evidenciam que, ao utilizar os PCs do RDE-AGMO, o controle consegue ajustar o nível de consumo com muito mais sucesso, obtendo uma redução no RMSE de 338. Isto se deve essencialmente a alta faixa de reduções alcançada pelas configurações do RDE-AGMO que o possibilitaram de ajustar o consumo da codificação conforme o necessário. O fato de o RDE-AGMO ter atingido erros menores para todos os casos de teste do SP70 afirmam a necessidade de uma melhor metodologia para a escolha de PCs de controladores assim como o potencial do RDE-AGMO de suprir tal necessidade.

O RMSE das atuações de um controle tem impacto direto na sua eficácia de atingir o consumo desejado com precisão. Esta precisão é confirmada pela Tabela 16 que exhibe as economias obtidas por ambos os controles para todos os casos de teste realizados. Assim como o esperado, o elevado erro obtido pelo CS de MACHADO (2017) para o vídeo *BQTerrace* no QP 22 o levou a ter uma redução 16,8% inferior ao esperado. Além disto, este vídeo obteve um erro médio na sua redução de 6,23%, o maior entre os vídeos de classe B. Por fim, ambos conjuntos de PCs obtiveram economias médias muito próximas do SP30 almejado, apresentando erros de apenas 3,54% e 5,73%.

Tabela 16 – Redução Energética Atingida por Ambos CSs para Todos Casos de Teste

Classe	Vídeo	QP	Economia de Energia (%)			
			SP30		SP70	
			(MACHADO, 2017)	RDE-AGMO	(MACHADO, 2017)	RDE-AGMO
C	BasketballDrill	22	32,28	32,12	16,79	67,09
		27	29,89	30,21	11,26	65,39
		32	31,92	32,80	11,93	66,71
		37	33,34	34,21	12,87	67,93
	BQMall	22	29,58	35,28	20,35	64,27
		27	28,65	29,67	17,53	61,39
		32	30,49	31,15	16,44	63,10
		37	32,33	31,22	16,58	65,35
	PartyScene	22	44,71	43,04	19,17	66,04
		27	42,74	51,99	15,62	64,66
		32	42,67	52,32	15,61	66,59
		37	44,75	50,84	15,65	69,62
B	BasketballDrive	22	16,15	35,85	15,11	70,04
		27	16,95	35,60	15,50	70,06
		32	18,03	35,54	17,46	70,32
		37	19,57	35,07	19,22	69,67
	BQTerrace	22	13,20	29,07	14,43	65,18
		27	14,76	36,83	16,32	69,29
		32	15,56	38,18	16,76	71,01
		37	16,81	40,83	17,96	72,11
	Cactus	22	15,11	35,27	16,04	68,18
		27	19,52	25,89	20,91	65,65
		32	20,20	25,48	22,95	65,82
		37	25,79	28,94	27,15	68,17
<b>Economia Média</b>			26,46	35,73	17,07	67,24

Já para o SP70 as economias atingidas diferem consideravelmente, com as configurações de MACHADO (2017) economizando 52,03% a menos do que o desejado, enquanto os PCs propostos pelo RDE-AGMO reduziram apenas 2,76% a menos do que o definido pelo SP70. Um comportamento interessante do controle de MACHADO (2017) de ser analisado é que este atingiu menores reduções ao tentar atingir o SP70 do que o SP30. Isto se deve pelo fato de o controle ter aplicado a sua configuração mais econômica ao longo de todas as codificações por não ter PCs com níveis de economia elevados o suficiente. Esta configuração por sua vez, acabou indesejadamente proporcionando reduções inferiores as demais para os vídeos testados. Este comportamento demonstra tanto a importância da heterogeneidade das economias dos PCs de um controlador quanto a imprecisão das metodologias atuais na definição desta heterogeneidade.

Outra padrão peculiar que pode ser observado para ambos os conjuntos de PCs é que estes exibiram uma alta dificuldade de controlar o consumo do vídeo *PartyScene*, apesar deste ter sido utilizado como base de avaliação para ambos CSs. Os PCs de MACHADO (2017) e do RDE-AGMO apresentaram erros médios de aproximadamente 14% e 20% respectivamente para este vídeo, sendo que o maior erro apresentado pelos CSs para os demais vídeos foi de 15% e de 6% para o vídeo *BQTerrace*.

Uma segunda característica importante de se avaliar em um controlador é o seu impacto em eficiência de codificação ao reduzir o consumo energético. A Tabela 17 mostra os BD-BRs causados por ambos CSs para cada um dos casos de teste avaliados.

Os PCs providenciados pelo RDE-AGMO aumentou a eficiência do controlador para o SP30 em 0,62%. Em vista que o BD-BR causado possui uma importância muito maior na performance do controle do que sua precisão na redução energética, podemos afirmar que este ganho é vantajoso mesmo com o acréscimo de 2,19% no erro de redução energética apresentada na Tabela 16.

Apesar de os vídeos *BQMall* e *BQTerrace* não terem apresentado altos RMSEs em ambos os conjuntos de PCs, estes apresentaram as maiores perdas em eficiência de codificação para ambos CSs. Estas altas perdas de eficiência de codificação são esperadas em vista que ambos vídeos obtiveram os piores BD-BRs para todas configurações encontradas nas suas avaliações exibidas pelas Tabelas 11 e 14. Ambos vídeos possuem objetos detalhados em movimento, como pessoas e carros, juntamente com uma movimentação constante da câmera. Este conjunto de características torna a estimação de movimento destes vídeos consideravelmente mais complexa, causando assim uma alta perda em eficiência de codificação sob configurações mais econômicas.

Tabela 17 – Comparação de BD-BRs dos Casos de Teste para Ambos CSs

Classe	Vídeo	BD-BR (%)			
		SP30		SP70	
		(MACHADO, 2017)	RDE-AGMO	MACHADO (2017)	RDE-AGMO
C	BasketballDrill	4,58	1,87	1,39	7,68
	BQMall	10,13	9,11	5	13,83
	PartyScene	5,69	2,66	2,47	7,42
B	BasketballDrive	0,77	1,74	0,73	7,53
	BQTerrace	1,74	2,93	1,86	7,75
	Cactus	1,53	2,36	1,54	7,6
<b>BD-BR Médio</b>		4,07	3,45	2,17	8,64
<b>Economia Média</b>		26,46	35,73	17,07	67,24

Embora a aplicação dos PCs encontrados pelo RDE-AGMO aumente o BD-BR em 6,54% com relação ao CS do *baseline*, isto só ocorre pelo fato de o controle do *baseline* falhar em atingir esse nível de economia e consumir 52% a mais do que o desejado. Além disto, com exceção do vídeo *BQMall* que causou um aumento discrepante de 13,83% no BD-BR, as perdas para os demais vídeos se mantiveram consideravelmente similares, estando todos próximos a um BD-BR de 7,6%.

Esta similaridade entre os BD-BRs do controlador ao utilizar o CS proposto levou o controlador a diminuir o desvio padrão do impacto causado entre os diferentes vídeos de 3,53% para 2,8%. Desta forma, pode-se afirmar que o RDE-AGMO além de providenciar uma melhor capacidade de atingir diferentes níveis de redução energética,

também tornou o controle mais eficiente e estável para cenários distintos.

Além do impacto em eficiência de codificação e a distância entre o SP e a economia obtida por um controlador, é necessário também estudarmos o impacto das configurações no comportamento deste controle. Pois trocas abruptas no consumo da codificação podem causar discrepâncias visuais de um momento para outro no vídeo sem necessidade. Assim sendo, é desejável que o controle altere o consumo da codificação o mais próximo possível do *SetPoint* da forma mais suave possível. Vale ressaltar também, que mudanças de cenas e características entre os quadros de um vídeo podem causar picos ou vales de consumo energético que o controle também deve ser capaz de lidar.

As Figuras 21 e 22 mostram o comportamento do consumo energético do x265 em torno do *SetPoint* 30 sob o controle de MACHADO (2017) utilizando o CS do *baseline* e o encontrado pelo RDE-AGMO ao longo dos GOPs da codificação do *BQMall* no QP 32. Vale lembrar que as atuações do controle sendo avaliado são realizadas a nível de GOPs de 4 quadros, o que resultará em um comportamento periódico no consumo do controlador proveniente da estrutura hierárquica do GOP. O vídeo *BQMall* neste QP específico foi selecionado para análise comportamental por ter providenciado RMSEs iguais para ambos PCs, o que nos permite um melhor estudo sobre o impacto das configurações nas flutuações de consumo energético causado pelo controle.

Ao analisarmos ambas figuras podemos notar que apesar de ambos conjuntos de configurações obterem RMSEs iguais e causarem picos de consumo, o comportamento do controlador se tornou bem mais estável ao utilizar os PCs encontrados neste trabalho. Enquanto *baseline* alterna entre configurações que consomem muito acima e abaixo do SP, com o RDE-AGMO é possível ajustar o consumo de forma mais suave em torno do SP.

Perceba também que houve quatro grandes picos de consumo mesmo ao se empregar os PCs encontrados pelo RDE-AGMO. Estes picos podem ser causados por características específicas do vídeo como pode ser observado no quadriculado vermelho em volta do GOP 43. Veja que nestas áreas há um leve aumento no consumo, seguido de uma pequena baixa e um grande pico no consumo para ambos CSs. Este comportamento específico nos mesmos quadros para ambos conjuntos torna evidente a influência de características da cena destes quadros no consumo energético da codificação.

As Figuras exibindo os comportamentos do controle para o SP 70 matêm comportamento similar para o conjunto de PCs propostos enquanto os PCs originais se matêm linearmente a cima do *SetPoint* por não conseguirem uma maior redução energética. Estes juntamente com os gráficos do comportamento do controle para os demais casos estudados podem ser encontrados nos Apêndice de A ao D.

Ao se codificar um vídeo em um dispositivo móvel, o nível de energia disponível

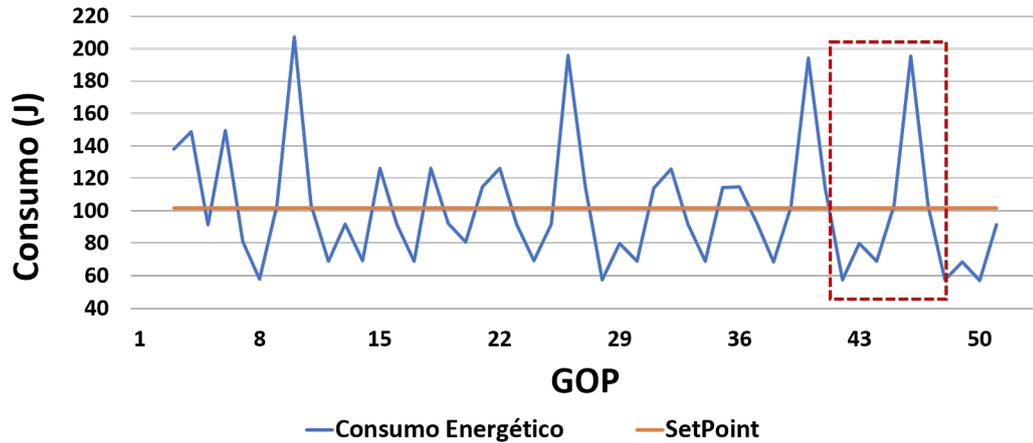


Figura 21 – Comportamento do Controle de MACHADO (2017) com os PCs Encontrados pelo RDE-AGMO (BasketballDrill QP 32 SP 30)

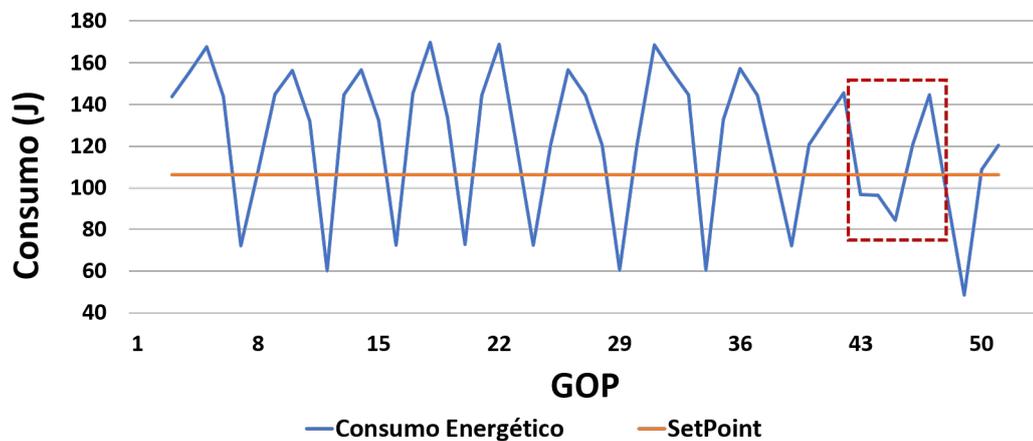


Figura 22 – Comportamento do Controle de MACHADO (2017) com os PCs Originais (BasketballDrill QP 32 SP 30)

em sua bateria varia constantemente, podendo assim mudar durante a codificação o quanto se deseja que ela consuma. Desta forma, um aspecto importante a ser levado em consideração ao se avaliar o comportamento de um controlador é a sua capacidade de adaptar o consumo durante a codificação caso o *SetPoint* seja alterado. Para simular este cenário, o vídeo *BasketballDrill* (QP 22) foi codificado sob o controle configurado para atingir uma redução inicial de 30%, sendo esta alterada para 60% após o vigésimo GOP. O comportamento do controle com ambos conjuntos de configurações pode ser observado nas Figuras 23 e 24.

Este experimento mostra que o controlador com os PCs do *baseline* não consegue atingir o *SetPoint* definido, com o consumo se distanciando ainda mais do desejado ao alterar o SP. Já ao inserir as configurações encontradas pelo RDE-AGMO, o controle conseguiu ajustar o consumo da codificação para o novo SP em apenas três GOPs, ou seja, apenas 12 quadros que seriam exibidos em 0,3s se considerarmos um vídeo de 30 fps. Os PCs também demonstraram maior estabilidade de controle para o SP60,

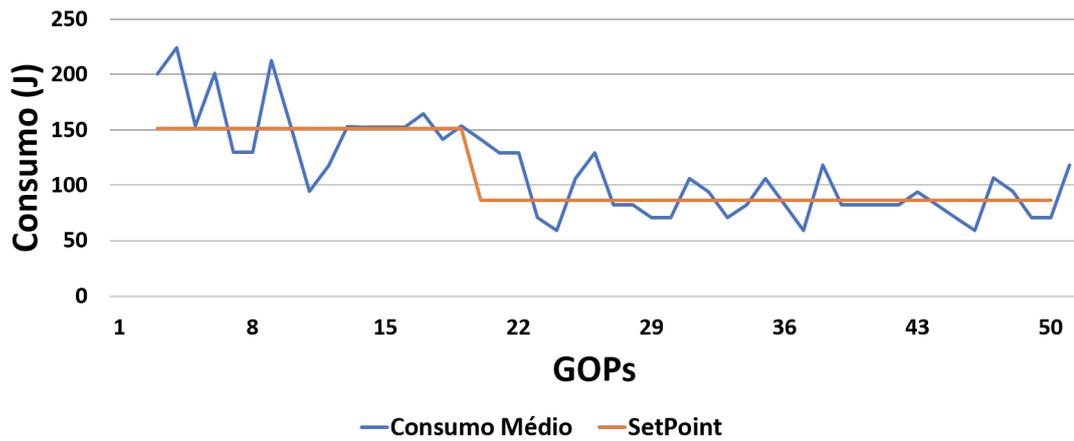


Figura 23 – Comportamento do Controle com os PCs Encontrados pelo RDE-AGMO sob Mudança de SP30 para SP60 (BasketballDrill QP 32)

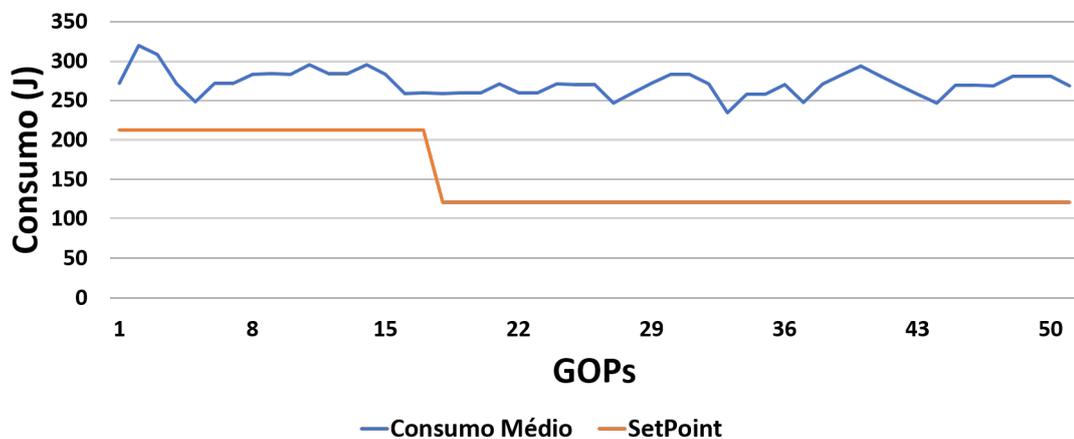


Figura 24 – Comportamento do Controle com os PCs Originais Sob Mudança de SP30 para SP60 (BasketballDrill QP 32)

apresentando maiores picos de consumo durante os 19 primeiros GOPs. Frente a este comportamento, os GOPs para o SP30 sofreram uma redução energética média de 32% que aumentou para 59% nos demais GOPs cujo SP foi alterado para 60%, apresentando assim erros médios de 2% e 1% para cada SP, respectivamente.

## 6 CONCLUSÕES E TRABALHOS FUTUROS

Esta dissertação apresentou um Algoritmo Genético Multiobjetivo capaz de selecionar automaticamente configurações eficientes de codificadores de vídeo HEVC. Ao cumprir tal tarefa o AGMO proposto denominado de RDE-AGMO permite a automação e aperfeiçoamento da análise de sensibilidade realizada no desenvolvimento de controladores para o padrão HEVC. Para desenvolver o RDE-AGMO foram necessários diversos estudos e experimentos para encontrar as técnicas mais apropriadas para o algoritmo.

A utilização de algoritmos genéticos para resolver problemas multiobjetivos da área de codificação de vídeo é uma inovação. Nas metodologias atualmente utilizadas para solucionar o problema de encontrar configurações eficientes na etapa de análise de sensibilidade, técnicas como frentes de Pareto e seleções empíricas são empregadas. O RDE-AGMO inova também o desenvolvimento de controladores para o padrão HEVC pois propõe o primeiro sistema de busca de configurações apropriado para o cenário multiobjetivo com inúmeras soluções.

A utilização de frentes de Pareto e seleções empíricas atualmente empregadas em outros trabalhos abrem espaço para melhorias no processo. Isto se deve por estas metodologias avaliarem um conjunto limitado de configurações, necessitarem de um especialista e demandarem um tempo considerável para sua execução. A complexidade da interdependência dos parâmetros de codificação do padrão HEVC fazem com que seja extremamente difícil encontrar as configurações mais eficientes possíveis analisando apenas algumas centenas de configurações. Além disto, esta busca fica extremamente dependente da aptidão e observação de um especialista, o que além de inserir possíveis falhas humanas ainda impossibilita a sua aplicação em cenários em que não se tem um especialista presente. Ademais, as diversas etapas e avaliações presentes nas atuais metodologias as levam a demandar um tempo considerável para serem executadas. Por fim, estas técnicas não permitem com que o desenvolvedor determine o quanto cada ponto de controle do seu controlador irá reduzir o seu objetivo.

O RDE-AGMO ao empregar os conceitos de algoritmos genéticos multiobjetivo

para encontrar configurações eficientes de codificadores HEVC, além de poder ser utilizado em qualquer codificador ou variável otimizável (energia, complexidade, etc.), também substitui as metodologias de análise de sensibilidade atualmente utilizadas no desenvolvimento de controladores. Ao ser empregado para automatizar a etapa de análise de sensibilidade no desenvolvimento de controladores, o RDE-AGMO foi capaz de diminuir o tempo deste processo da escala de meses para dias e isentou a necessidade de um especialista para definir e avaliar os PCs sendo definidos. Além disto, o algoritmo também permitiu que a distribuição de economia entre os PCs do controlador seja definida com precisão.

O algoritmo desenvolvido neste trabalho foi fruto de um extensivo estudo de diversas técnicas e metodologias de AGMOs para os diversos operadores genéticos que o constituem. Este estudo permitiu com que fossem encontrados operadores apropriados para o cenário deste trabalho, permitindo proporcionar ao RDE-AGMO uma alta eficiência e boa velocidade de convergência tanto em relação ao RD quanto ao consumo.

Ao avaliar aproximadamente 4 mil configurações do codificador x265, nossos experimentos encontraram 9 configurações energeticamente eficientes para vídeos 448p e 4 configurações para vídeos 1080p. Estes dois conjuntos de configurações denominados de CSs possuem PCs com distribuições de economia muito próximas do desejado, apresentando reduções nos MSEs médios nas suas economias de até 61% para o primeiro CS (448p) e de 95% para o segundo conjunto de PCs (1080p) quanto comparados com os CSs dos demais controladores. Esta redução significativa no MSE da distribuição das reduções de cada PC salienta a competência do RDE-AGMO em definir o consumo da configuração que esta buscando. A generalidade da aplicação do RDE-AGMO para múltiplos vídeos foi constatada perante as perdas médias de eficiência de codificação causadas serem similares para todos casos de teste.

Por fim, para evidenciar o impacto da utilização do RDE-AGMO na realização da análise de sensibilidade no desenvolvimentos de controladores, o controle de energia desenvolvido por MACHADO (2017) foi utilizado para codificar múltiplos vídeos com o seus PCs originais e os definidos neste trabalho. Nesta comparação comprovou-se que ao substituir as atuais análises de sensibilidade pelo RDE-AGMO o controlador se tornou capaz de atingir SPs de até 70%, o que não conseguira com o seu CS do *baseline*. Além disto, diminui o impacto em BD-BR de 4,07% para 3,45% em troca de um erro de apenas 2,19% no controle do seu consumo. Ainda, o comportamento energético da codificação se tornou mais estável e capaz de se adaptar a mudanças de SP com erros médios de apenas 2% no consumo dos GOPs. Por fim, o controle se mostrou capaz de reagir rapidamente à trocas de *SetPoint* ao longo da codificação.

## 6.1 Trabalhos Futuros

O AGMO proposto por este trabalho traz uma solução eficaz e genérica para a obtenção de configurações eficientes de codificadores HEVC. Frente a isto, o RDE-AGMO pode ser aplicado a qualquer codificador ou variável do padrão HEVC. Logo, há uma grande gama de cenários e experimentos que podem ser feitos com codificadores deste padrão. Por exemplo, poderia-se utilizar o RDE-AGMO para substituir as análises de complexidade aritmética e temporal realizadas nos controladores de GRELLERT (2014) e CORREA et al. (2015) com a finalidade de tentar melhorar a eficiência destes controles. Ou ainda encontrar configurações que impactem positivamente o processo de decodificação de vídeo, usando o princípio *coding for decoding*, para assim reduzir energia no decodificador. Poderia-se também aplicar o algoritmo para descobrir taxas de imprecisão de memória que determinados módulos do codificador podem tolerar, para assim se atingir economias de energia através de conceitos de armazenamento aproximado sem gerar grandes perdas na eficiência da codificação. Vale ressaltar também, que todos os possíveis estudos citados podem ser replicados para qualquer codificador do padrão HEVC.

Como trabalhos futuros caberia também realizar experimentos com o objetivo de aperfeiçoar certos aspectos do RDE-AGMO para que este atinja configurações ainda mais eficientes. Entre as possíveis melhoras podemos citar a utilização de diferentes vídeos concatenados como função de avaliação, para assim tornar as configurações mais genéricas e, conseqüentemente, diminuir a diferença nos BD-BRs das configurações para vídeos com características distintas. Ainda focando na função de avaliação, utilizar codificar os quatro QPs definidos pelas CTCs e utilizados no cálculo do BD-BR reduziria consideravelmente o impacto em eficiência de codificação das configurações encontradas. Por fim, caberia também analisar os benefícios de aumentar o número de gerações utilizadas ao se aplicar o RDE-AGMO para vídeos com resolução *Full HD*.

## REFERÊNCIAS

ABD-ALSABOUR, N. A review on evolutionary feature selection. In: EUROPEAN MODELLING SYMPOSIUM, 2014., 2014. **Anais...** [S.l.: s.n.], 2014. p.20–26.

BJØNTEGAARD, G. Improvements of the BD-PSNR model. **ITU-T**, [S.l.], 2008.

BOYCE, J. HM16: High Efficiency Video Coding Test Model (HM16) Encoder Description. **JCTVC-R1002**, Sapporo, 2014.

BROSS, B. et al. High Efficiency Video Coding (HEVC) text specification Working Draft 5. **JCT-VC**, [S.l.], 2011.

CERRADA, M. et al. Fault diagnosis in spur gears based on genetic algorithm and random forest. **Mechanical Systems and Signal Processing**, [S.l.], v.70, p.87–103, 2016.

CISCO. **Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update**. Disponível em: <<http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html>>. Acesso em: 20 Janeiro 2020.

CORREA, G.; ASSUNCAO, P. A.; AGOSTINI, L. V.; SILVA CRUZ, L. A. da. Fast HEVC encoding decisions using data mining. **IEEE transactions on circuits and systems for video technology**, [S.l.], v.25, n.4, p.660–673, 2015.

CORRÊA, G. R. Pareto-Based Method for High Efficiency Video Coding With Limited Encoding Time Published. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.26, p.1734–1745, Sept 2016.

CORRÊA, G. R. et al. Performance and Computational Complexity Assessment of High-Efficiency Video Encoders. **Circuits and Systems for Video Technology, IEE Transactions on**, [S.l.], v.22, n.12, p.1899–1909, 2012.

DARWIN, C. **On the Origin of Species by Means of Natural Selection (J. Murray, London)**.

DEB, K. Multi-objective optimization. In: **Search methodologies**. [S.l.]: Springer, 2014. p.403–449.

FONSECA, C. M.; FLEMING, P. J. Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation. **IEEE Transactions on Systems, Man, and Cybernetics**, [S.l.], v.28, n.1, p.26–37, 1998.

GHANBARI, M. **Standard Codecs: Image Compression to Advanced Video Coding**. United Kingdom: The Institution of Electrical Engineers, 2003.

GOLDBERG, D. E. Genetic algorithms in search. **Optimization, and Machine Learning**, [S.l.], 1989.

GONZALEZ R., W. R. **Processamento de Imagens Digitais**. São Paulo: Blücher, 2003. v.3.

GOSH, A. Evolutionary Algorithms for Multi-Criterion Optimization: A Survey. **International Journal of Computing & Information Sciences**, [S.l.], 2004.

GRELLERT, M. **Computational effort analysis and control in High Efficiency Video Coding**. 2014. Dissertação — Universidade Federal do Rio Grande do Sul, Porto Alegre.

GROUP, U. V. **Kvazaar Encoder**. Disponível em: <<http://ultravideo.cs.tut.fi/>>. Acesso em: 20 Janeiro 2020.

HOLLAND, J. H. et al. **Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence**. [S.l.]: MIT press, 1992.

HORN, J. rey; NAFPLIOTIS, N.; GOLDBERG, D. E. A niched Pareto genetic algorithm for multiobjective optimization. In: IEEE CONFERENCE ON EVOLUTIONARY COMPUTATION, IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE, 1994. **Proceedings...** [S.l.: s.n.], 1994. v.1, p.82–87.

ITU-T. ITU-T Recommendation H.265: High Efficiency Video Coding. **ITU-T.**, [S.l.], 2015.

KAYA, Y.; UYAR, M. et al. A novel crossover operator for genetic algorithms: ring crossover. **arXiv preprint arXiv:1105.0355**, [S.l.], 2011.

MACHADO, I. D. **Controlador de Energia para o Codificador x265 Utilizando Frontes de Pareto**. 2017. Trabalho de Conclusão — Universidade Federal de Pelotas, Pelotas.

MAGALHAES-MENDES, J. A comparative study of crossover operators for genetic algorithms to solve the job shop scheduling problem. **WSEAS transactions on computers**, [S.l.], v.12, n.4, p.164–173, 2013.

METAWA, N.; HASSAN, M. K.; ELHOSENY, M. Genetic algorithm based model for optimizing bank lending decisions. **Expert Systems with Applications**, [S.l.], v.80, p.75–82, 2017.

MONTEIRO, E.; GRELLERT, M.; ZATT, B.; BAMPI, S. Rate-distortion and energy performance of HEVC video encoders. In: INTERNATIONAL WORKSHOP ON POWER AND TIMING MODELING, OPTIMIZATION AND SIMULATION (PATMOS), 2014., 2014. **Anais...** [S.l.: s.n.], 2014. p.1–8.

MULTICOREWARE. **x265**. Disponível em: <<http://x265.org/>>. Acesso em: 15 Março 2017.

OCHOA, G.; HARVEY, I.; BUXTON, H. Optimal mutation rates and selection pressure in genetic algorithms. In: ANNUAL CONFERENCE ON GENETIC AND EVOLUTIONARY COMPUTATION, 2., 2000. **Proceedings...** [S.l.: s.n.], 2000. p.315–322.

OGATA, K.; YANG, Y. **Modern control engineering**. [S.l.]: Prentice-Hall, 2002. v.4.

PAN, J. **RAPL (Running Average Power Limit) driver**. Disponível em: <<http://lwn.net/Articles/545745/>>. Acesso em: 15 Março 2017.

PENNY, W. I. **Pareto-based energy control for the HEVC encoder**. 2017. Dissertação — , Pelotas.

QIU, M. et al. Phase-change memory optimization for green cloud with genetic algorithm. **IEEE Transactions on Computers**, [S.l.], v.64, n.12, p.3528–3540, 2015.

RICHARDSON, I. **Video Codec Design: Developing Image and Video Compression Systems**. Chichester: John Wiley and Sons, 2002.

ROBYR, J.-L.; GONON, F.; FAVRE, L.; NIEDERHÄUSER, E.-L. Convergence of multi-criteria optimization of a building energetic resources by genetic algorithm. In: INTERNATIONAL CONFERENCE ON SMART GRID AND CLEAN ENERGY TECHNOLOGIES (ICSGCE), 29 MAY-1 JUNE 2018, KAJANG, MALAYSIA, 2018., 2018. **Proceedings...** [S.l.: s.n.], 2018. n.CONFERENCE.

ROSEWARNE, C.; BROSS, B.; SHARMAN, K.; SULLIVAN, G. JCTVCU1002: High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Improved Encoder Description. **Tech. Rep., Joint Collaborative Team on Video Coding (JCT-VC)**, [S.l.], 2015.

SILVA, E. E. d. **Otimização de estruturas de concreto armado utilizando algoritmos genéticos**. 2001. Tese (Doutorado em Ciência da Computação) — Universidade de São Paulo.

SOARES, G. L. Algoritmos genéticos: estudo, novas técnicas e aplicações. **Belo Horizonte: Universidade Federal de Minas Gerais**, [S.l.], p.40, 1997.

SRINIVAS, M.; PATNAIK, L. M. Genetic algorithms: A survey. **computer**, [S.l.], v.27, n.6, p.17–26, 1994.

SUKSTRIENWONG, A. A Price-Based Mechanism for Online Buyer Coalition by Genetic Algorithms. **International Journal of Innovative Computing, Information and Control (IJICIC)**, [S.l.], v.14, n.05, p.1653–1679, 2018.

SULLIVAN, G. J. et al. Overview of the High Efficiency Video Coding (HEVC) Standard. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.22, n.12, p.882 – 885, may 2012.

SZE, V. e. a. **High Efficiency Video Coding (HEVC): Algorithms and Architectures**. New York: Springer, 2014.

TABASSUM, M.; MATHEW, K. et al. A genetic algorithm analysis towards optimization solutions. **International Journal of Digital Information and Wireless Communications (IJDIWC)**, [S.l.], v.4, n.1, p.124–142, 2014.

VIKHAR, P. A. Evolutionary algorithms: A critical review and its future prospects. In: INTERNATIONAL CONFERENCE ON GLOBAL TRENDS IN SIGNAL PROCESSING, INFORMATION COMPUTING AND COMMUNICATION (ICGTSPICC), 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p.261–265.

WILLIS, M. Proportional-integral-derivative control. **Dept. of Chemical and Process Engineering University of Newcastle**, [S.l.], 1999.

## APÊNDICE A COMPORTAMENTO DO CONTROLE COM AS CONFIGURAÇÕES ENCONTRADAS VIDEOS CLASSE C SP30

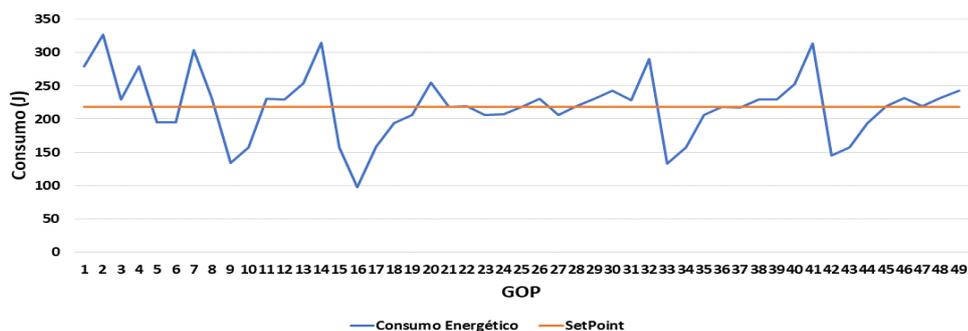


Figura A.1: Comportamento do Controle para o Vídeo BasketballDrill (QP - 22)

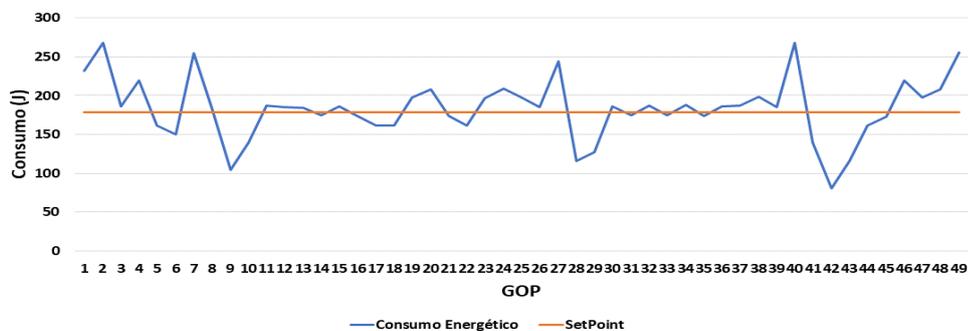


Figura A.2: Comportamento do Controle para o Vídeo BasketballDrill (QP - 27)

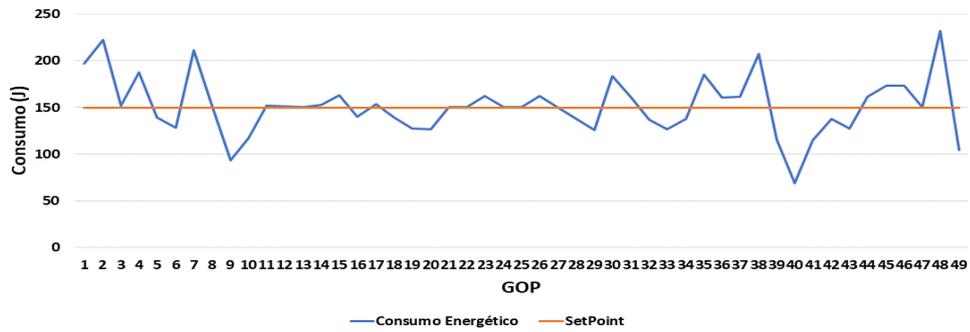


Figura A.3: Comportamento do Controle para o Vídeo BasketballDrill (QP - 32)

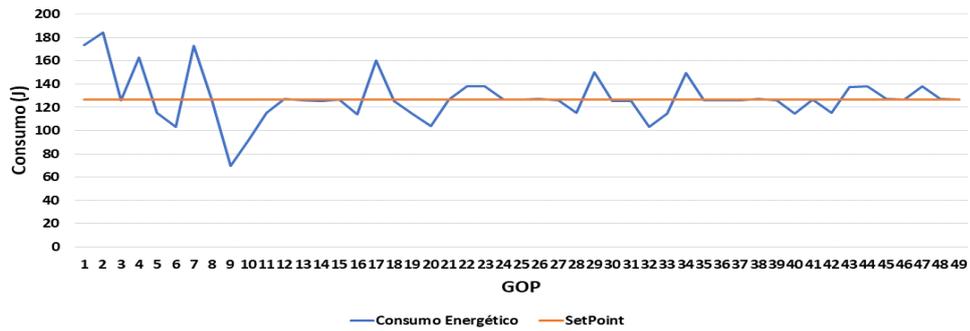


Figura A.4: Comportamento do Controle para o vídeo BasketballDrill (QP - 37)

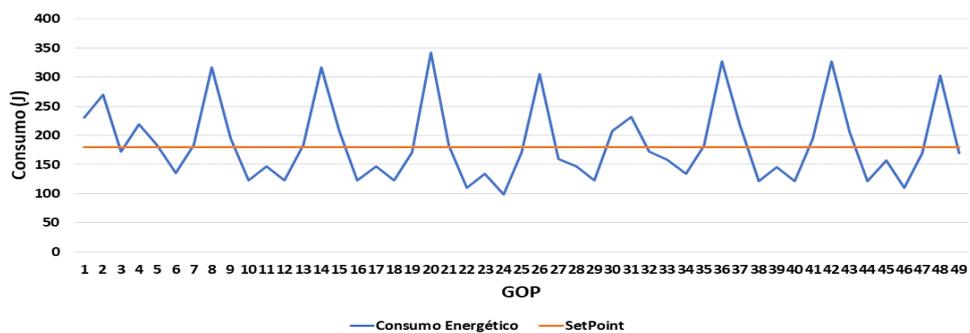


Figura A.5: Comportamento do Controle para o Vídeo BQMall (QP - 22)

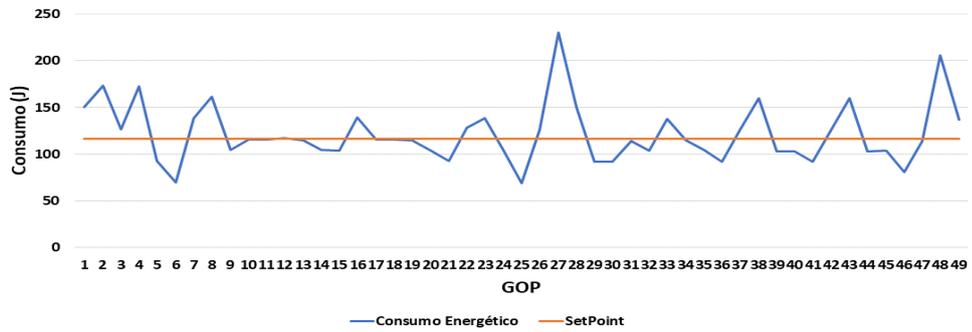


Figura A.6: Comportamento do Controle para o Vídeo BQMall (QP - 27)

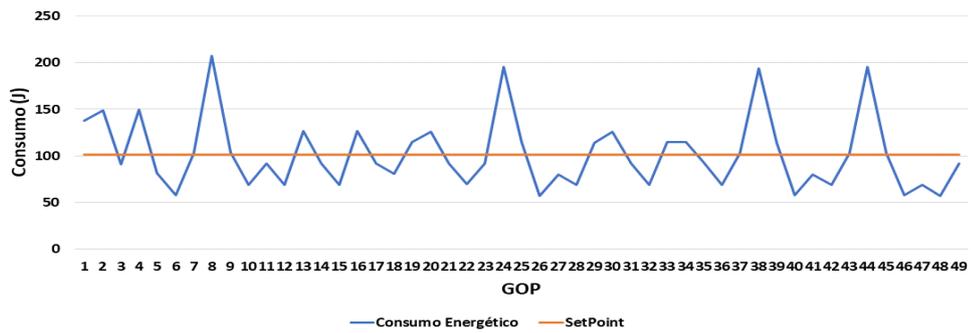


Figura A.7: Comportamento do Controle para o Vídeo BQMall (QP - 32)

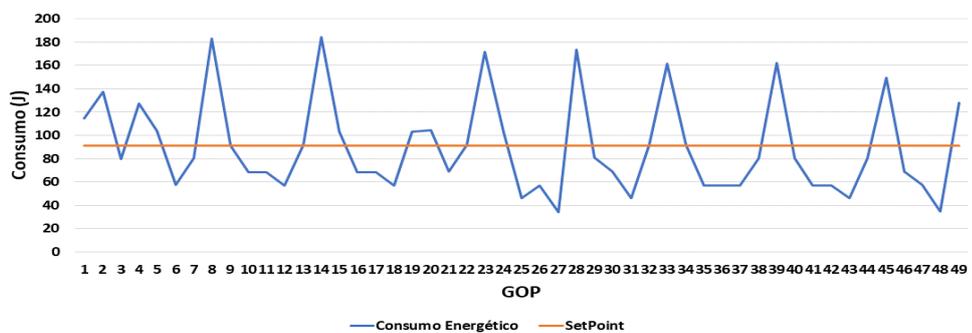


Figura A.8: Comportamento do Controle para o vídeo BQMall (QP - 37)

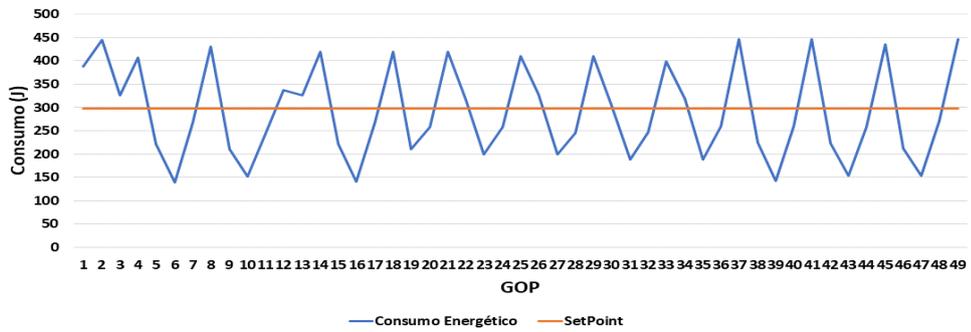


Figura A.9: Comportamento do Controle para o Vídeo PartyScene (QP - 22)

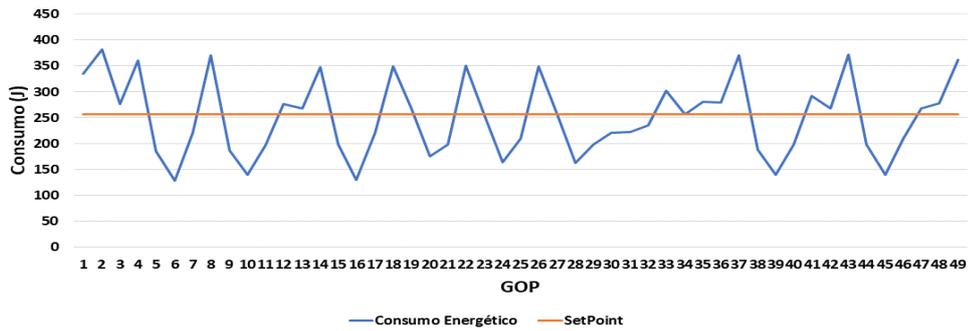


Figura A.10: Comportamento do Controle para o Vídeo PartyScene (QP - 27)

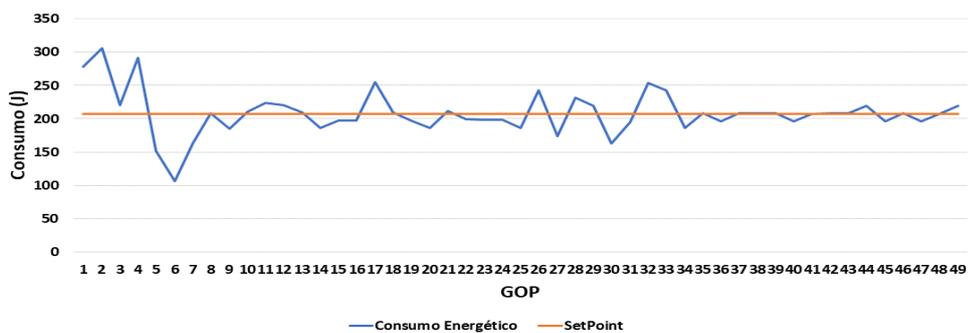


Figura A.11: Comportamento do Controle para o Vídeo PartyScene (QP - 32)

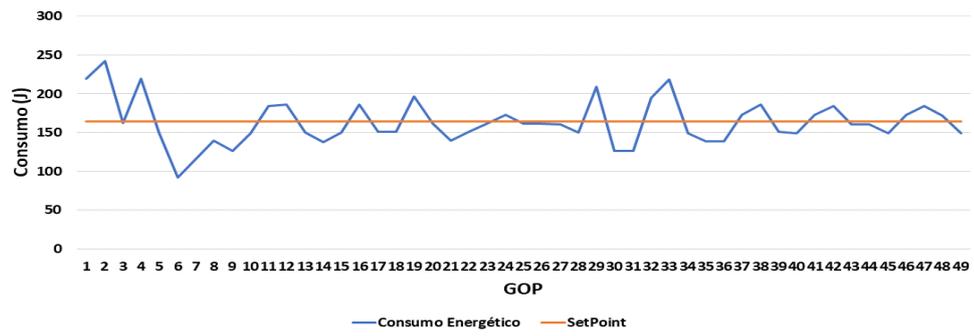


Figura A.12: Comportamento do Controle para o vídeo PartyScene (QP - 37)

## APÊNDICE B COMPORTAMENTO DO CONTROLE COM AS CONFIGURAÇÕES ENCONTRADAS VIDEOS CLASSE C SP70

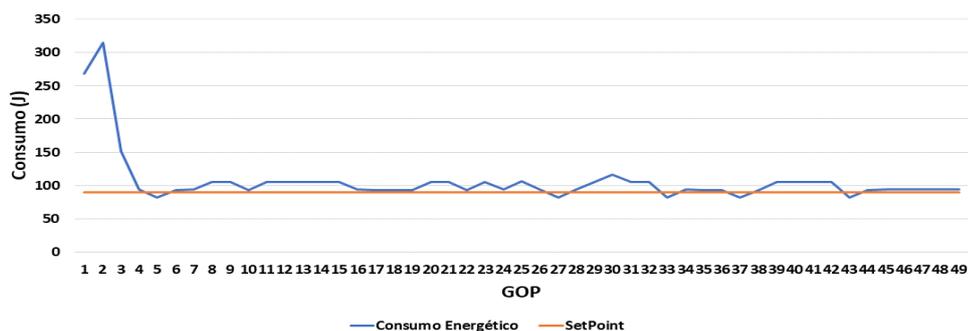


Figura B.1: Comportamento do Controle para o Vídeo BasketballDrill (QP - 22)

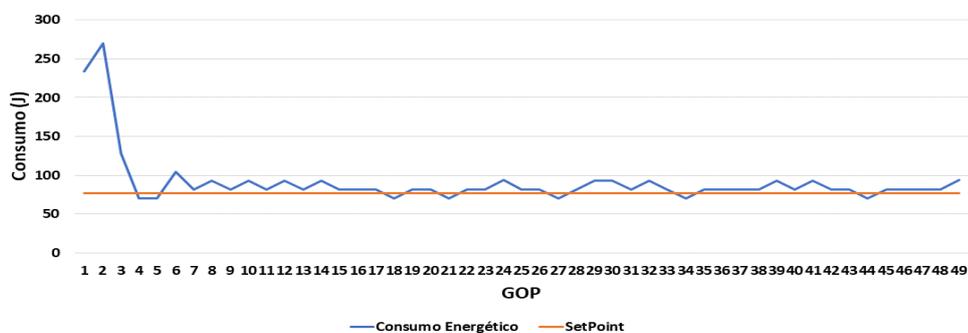


Figura B.2: Comportamento do Controle para o Vídeo BasketballDrill (QP - 27)

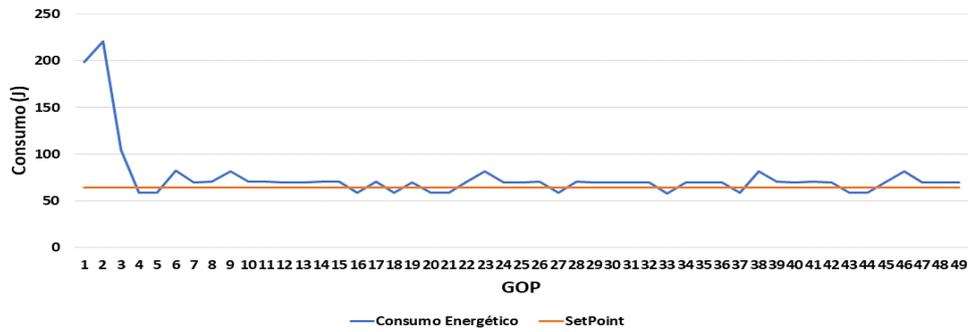


Figura B.3: Comportamento do Controle para o Vídeo BasketballDrill (QP - 32)

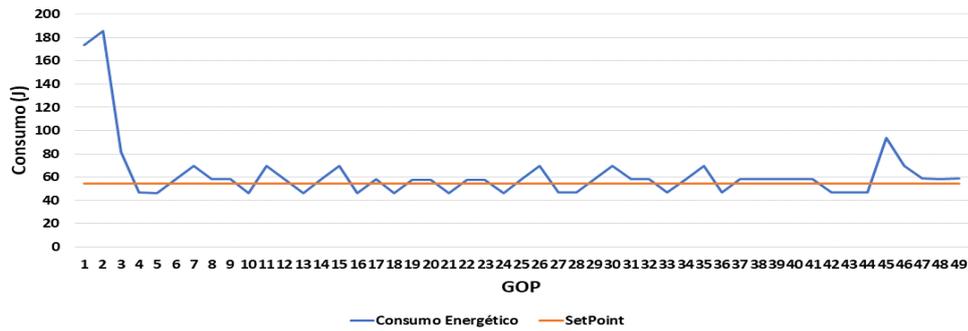


Figura B.4: Comportamento do Controle para o vídeo BasketballDrill (QP - 37)

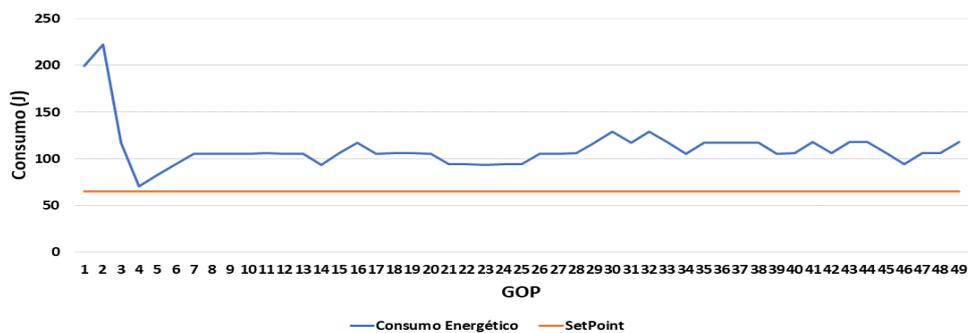


Figura B.5: Comportamento do Controle para o Vídeo BQMall (QP - 22)

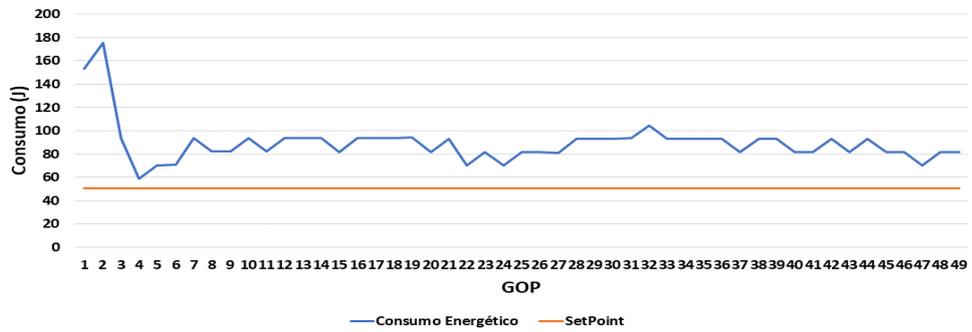


Figura B.6: Comportamento do Controle para o Vídeo BQMall (QP - 27)

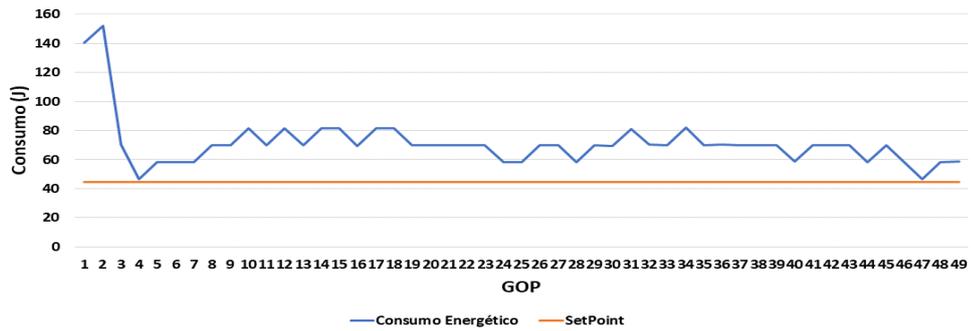


Figura B.7: Comportamento do Controle para o Vídeo BQMall (QP - 32)

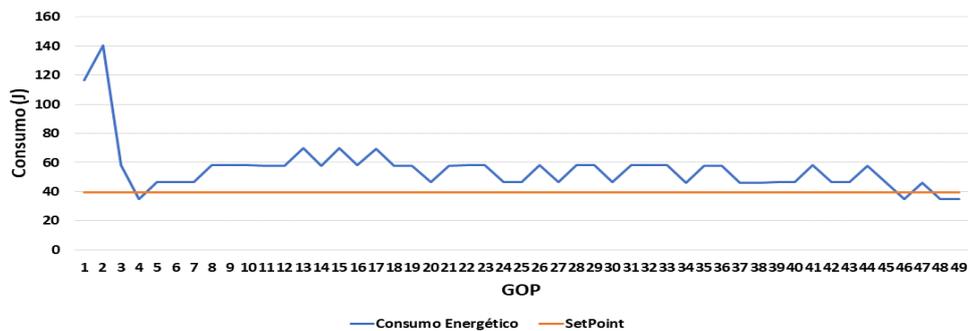


Figura B.8: Comportamento do Controle para o vídeo BQMall (QP - 37)

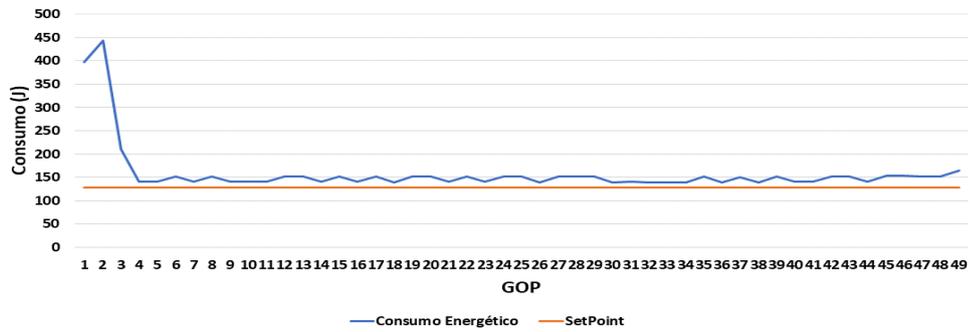


Figura B.9: Comportamento do Controle para o Vídeo PartyScene (QP - 22)

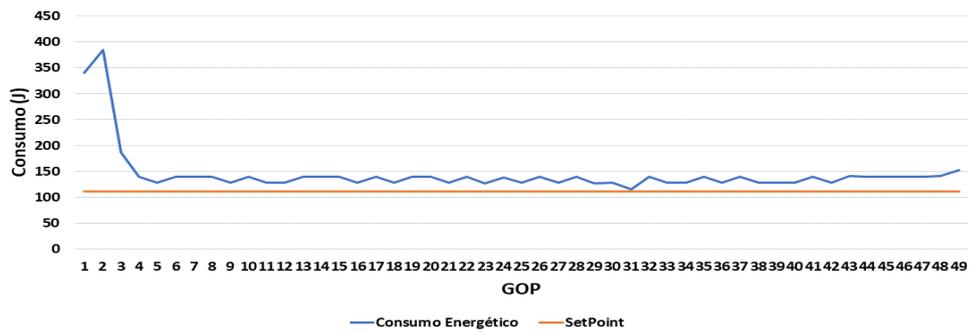


Figura B.10: Comportamento do Controle para o Vídeo PartyScene (QP - 27)

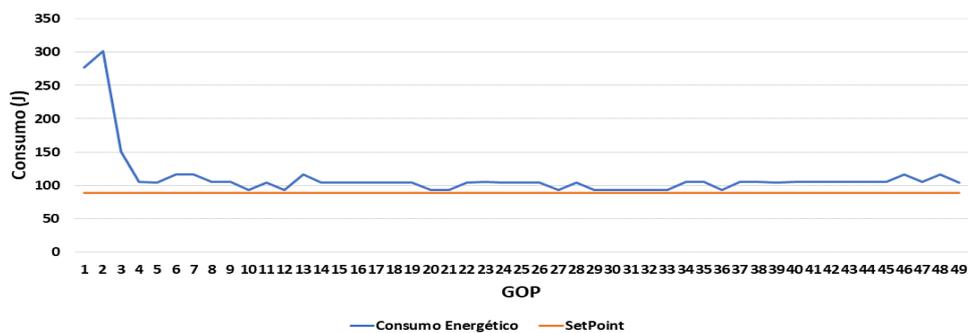


Figura B.11: Comportamento do Controle para o Vídeo PartyScene (QP - 32)

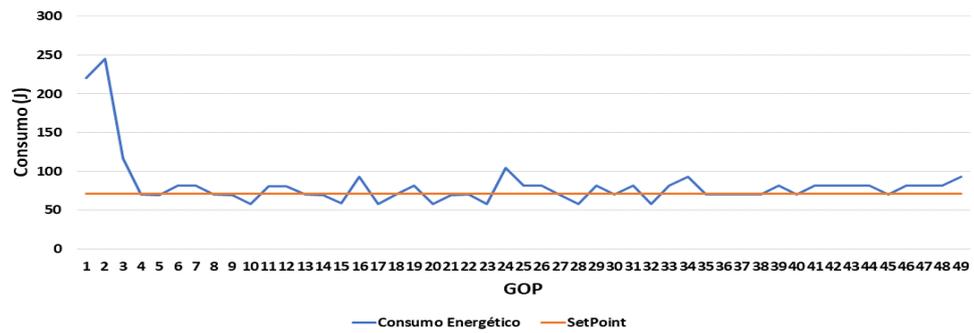


Figura B.12: Comportamento do Controle para o vídeo PartyScene (QP - 37)

## APÊNDICE C COMPORTAMENTO DO CONTROLE COM AS CONFIGURAÇÕES ENCONTRADAS VIDEOS CLASSE B SP30

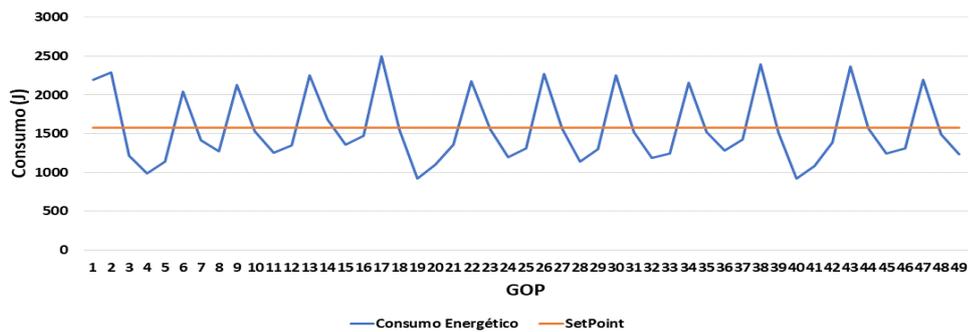


Figura C.1: Comportamento do Controle para o Vídeo BasketballDrive (QP - 22)

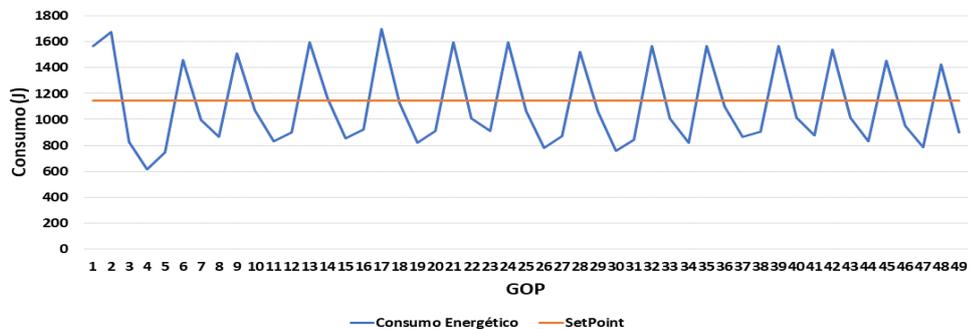


Figura C.2: Comportamento do Controle para o Vídeo BasketballDrive (QP - 27)

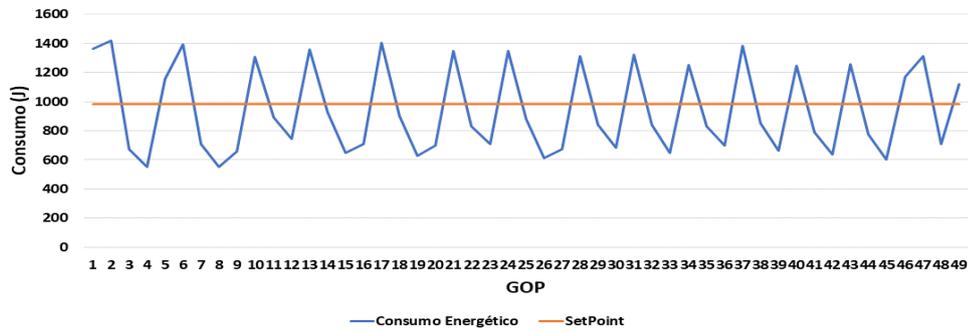


Figura C.3: Comportamento do Controle para o Vídeo BasketballDrive (QP - 32)

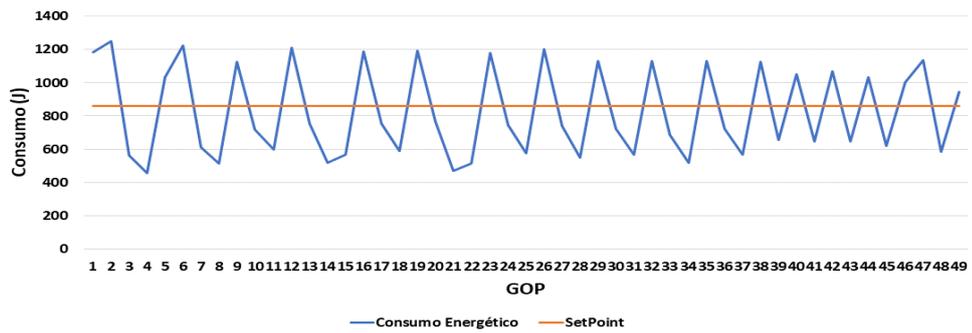


Figura C.4: Comportamento do Controle para o vídeo BasketballDrive (QP - 37)

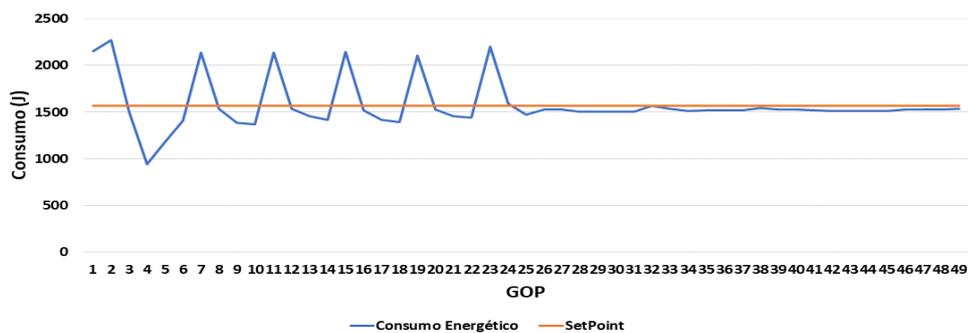


Figura C.5: Comportamento do Controle para o Vídeo BQTerrace (QP - 22)

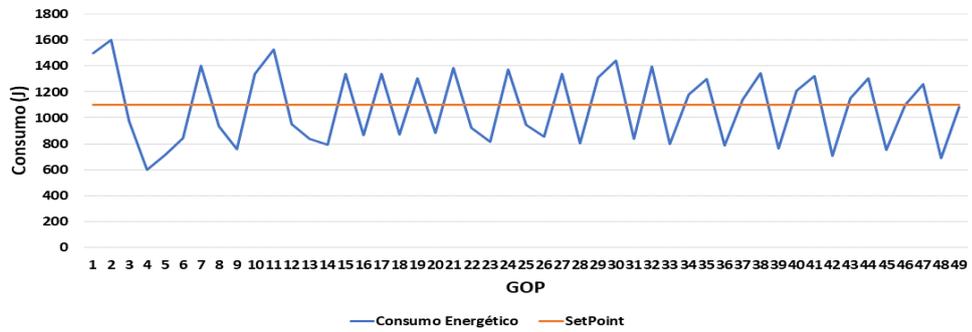


Figura C.6: Comportamento do Controle para o Vídeo BQTerrace (QP - 27)

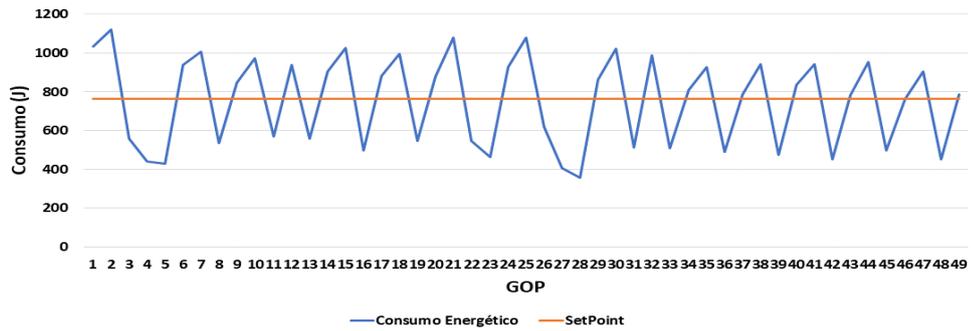


Figura C.7: Comportamento do Controle para o Vídeo BQTerrace (QP - 32)

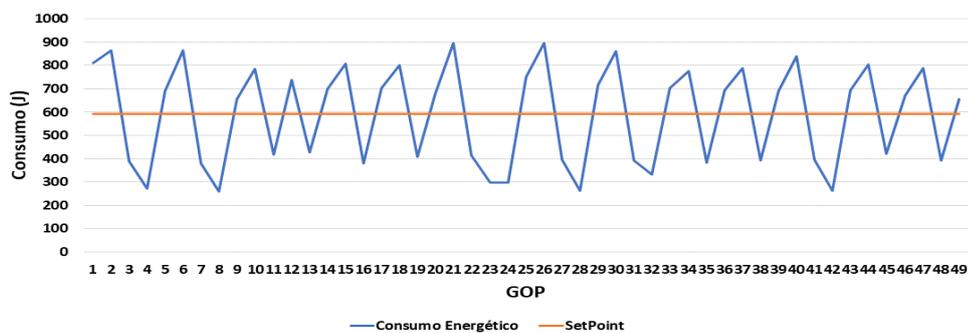


Figura C.8: Comportamento do Controle para o vídeo BQTerrace (QP - 37)

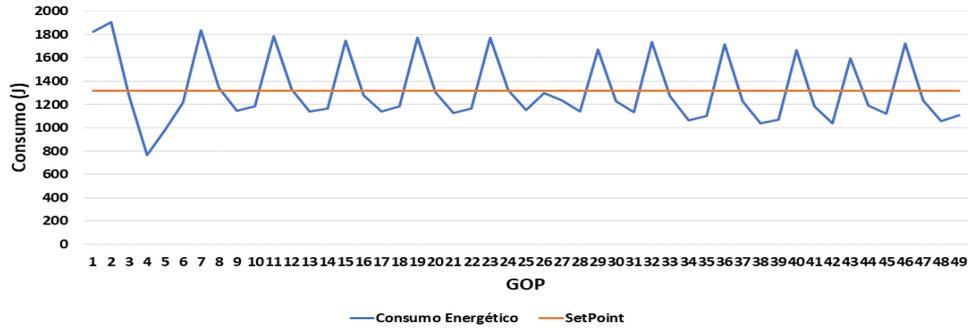


Figura C.9: Comportamento do Controle para o Vídeo Cactus (QP - 22)

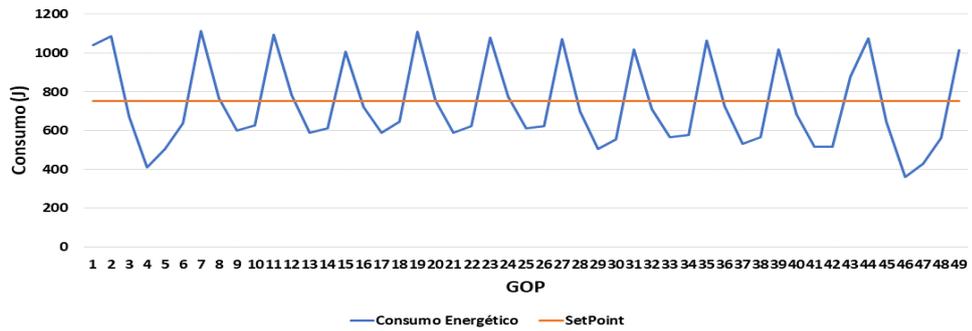


Figura C.10: Comportamento do Controle para o Vídeo Cactus (QP - 27)

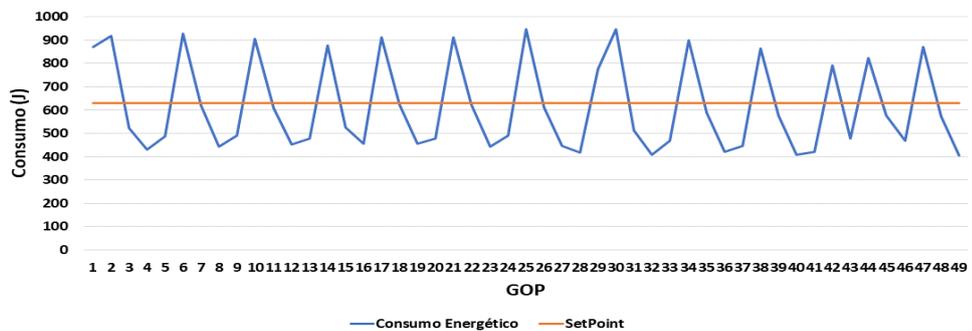


Figura C.11: Comportamento do Controle para o Vídeo Cactus (QP - 32)

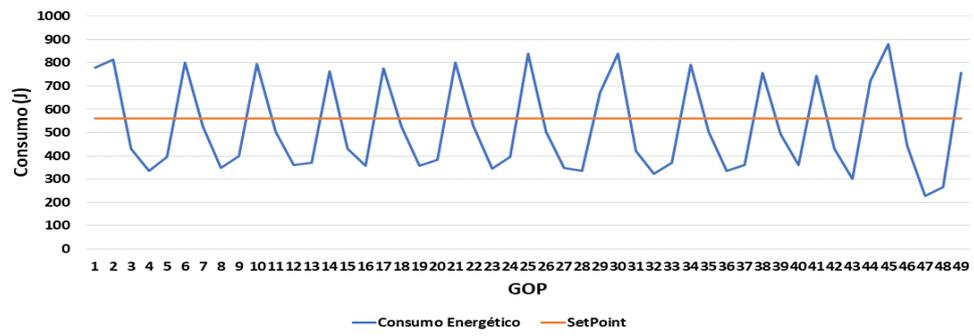


Figura C.12: Comportamento do Controle para o vídeo Cactus (QP - 37)

## APÊNDICE D COMPORTAMENTO DO CONTROLE COM AS CONFIGURAÇÕES ENCONTRADAS VIDEOS CLASSE B SP70

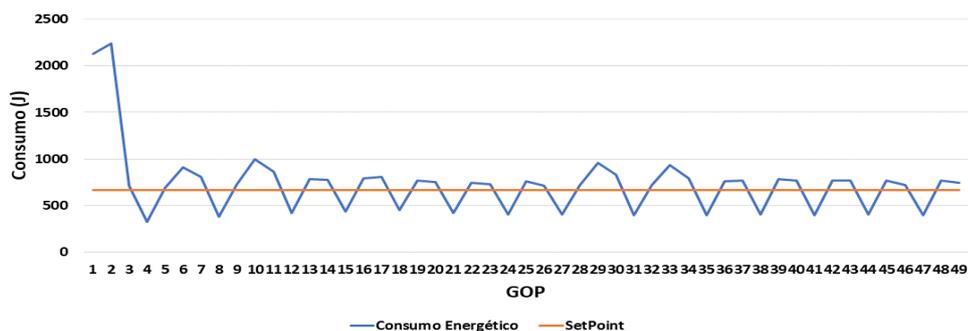


Figura D.1: Comportamento do Controle para o Vídeo BasketballDrive (QP - 22)

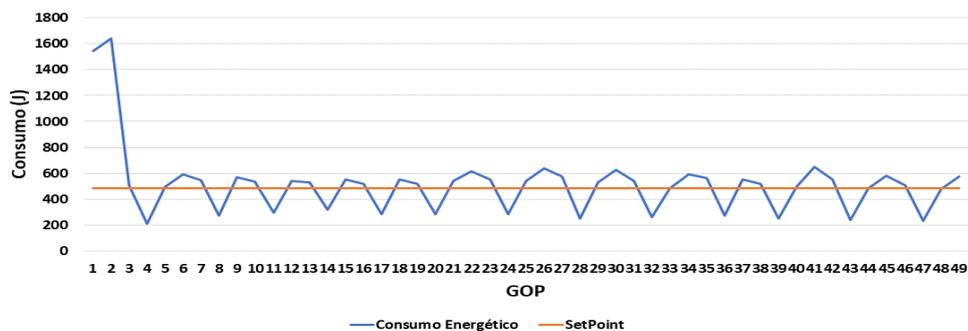


Figura D.2: Comportamento do Controle para o Vídeo BasketballDrive (QP - 27)

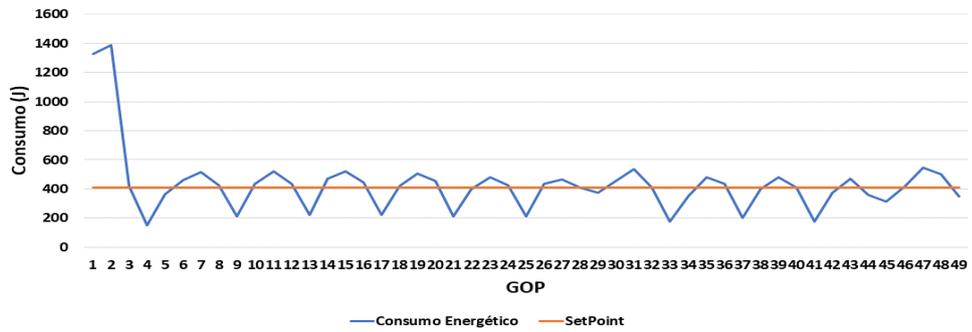


Figura D.3: Comportamento do Controle para o Vídeo BasketballDrive (QP - 32)

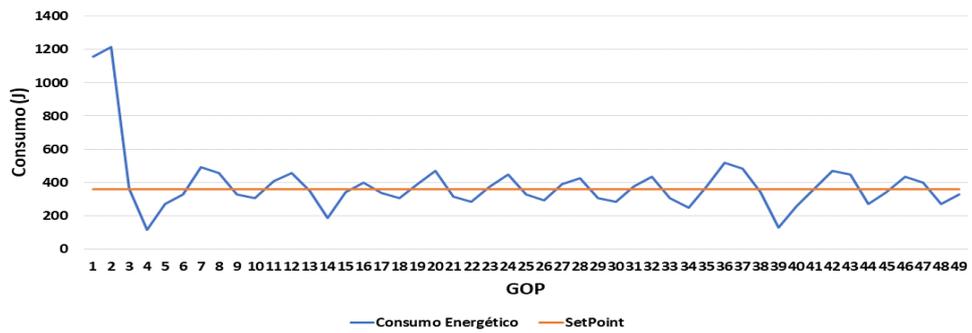


Figura D.4: Comportamento do Controle para o vídeo BasketballDrive (QP - 37)

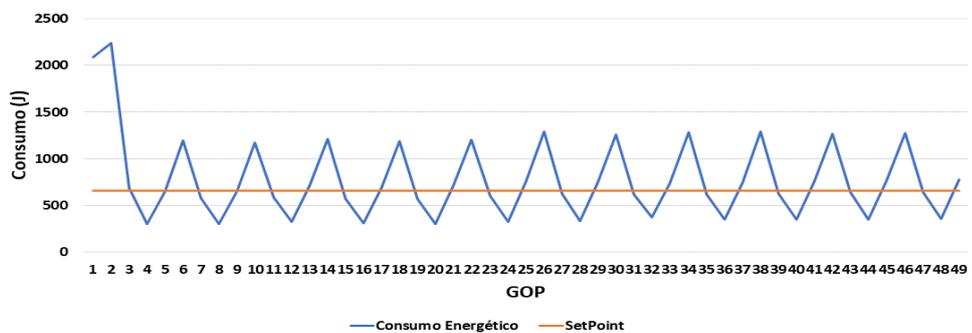


Figura D.5: Comportamento do Controle para o Vídeo BQTerrace (QP - 22)

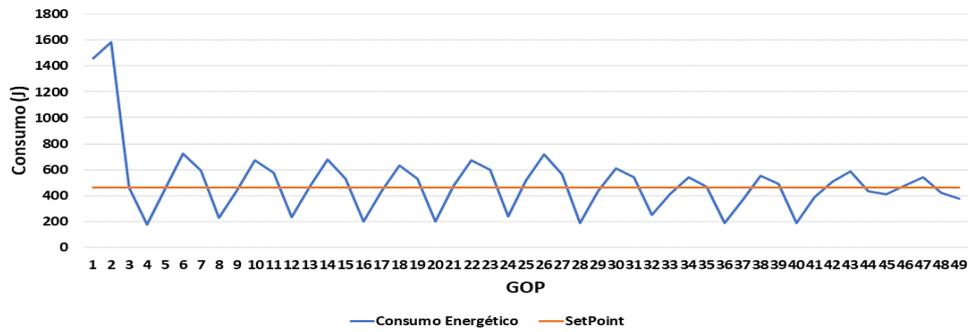


Figura D.6: Comportamento do Controle para o Vídeo BQTerrace (QP - 27)

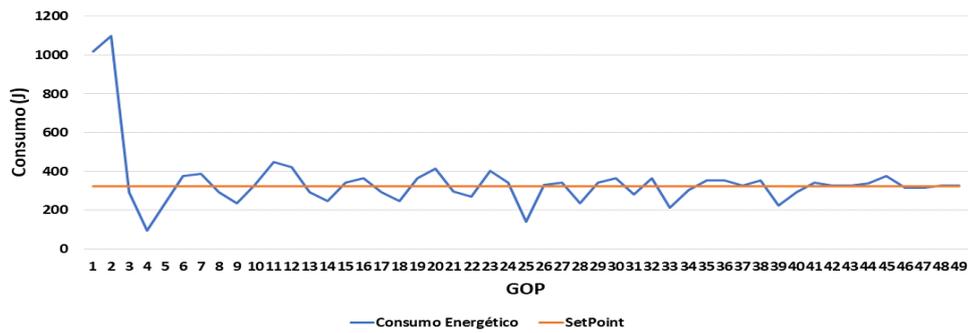


Figura D.7: Comportamento do Controle para o Vídeo BQTerrace (QP - 32)

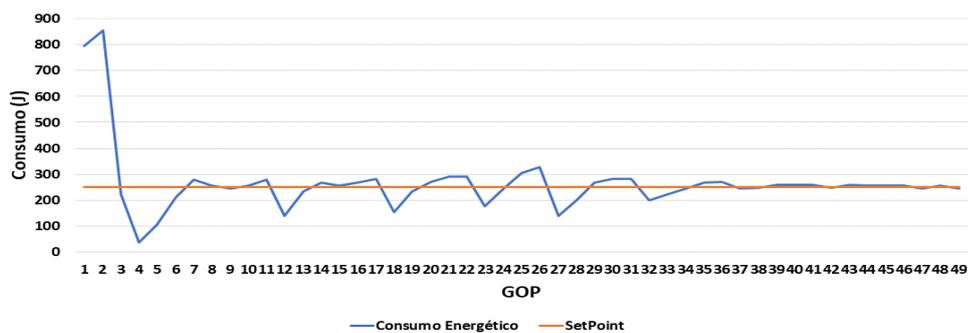


Figura D.8: Comportamento do Controle para o vídeo BQTerrace (QP - 37)

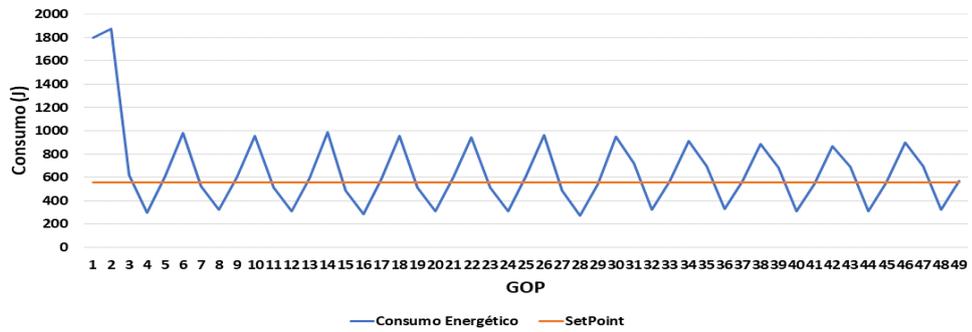


Figura D.9: Comportamento do Controle para o Vídeo Cactus (QP - 22)

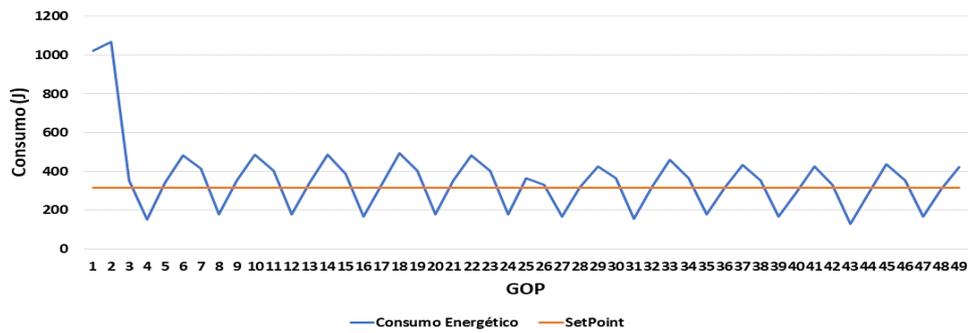


Figura D.10: Comportamento do Controle para o Vídeo Cactus (QP - 27)

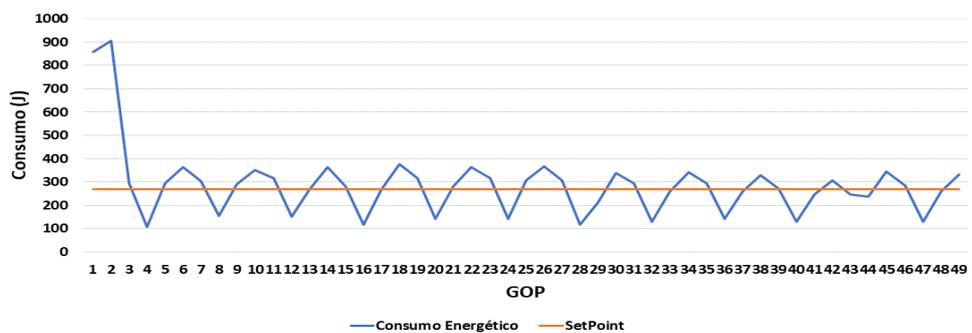


Figura D.11: Comportamento do Controle para o Vídeo Cactus (QP - 32)

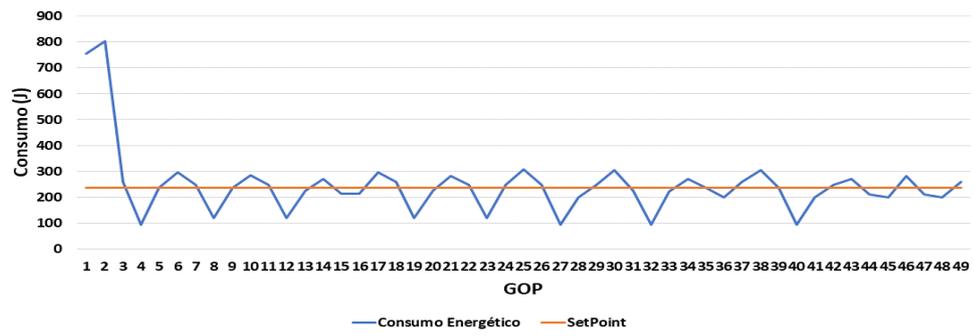


Figura D.12: Comportamento do Controle para o vídeo Cactus (QP - 37)