UNIVERSIDADE FEDERAL DA BAHIA

COLEGIADO DE ENGENHARIA DE COMPUITAÇÃO

AVALIAÇÃO EXPERIMENTAO – HUFFMAN ADAPTATIVO

DANIEL DO NASCIMENTO GOMES

SALVADOR

2023

Introdução

A Codificação de Huffman Adaptativo é uma técnica relacionada ao método de Huffman, usado para comprimir dados. A principal diferença é que, enquanto o método de Huffman padrão requer conhecimento prévio das frequências de ocorrência dos símbolos, o método adaptativo lida com casos em que essas frequências não são conhecidas com antecedência.

Motivos da Escolha

1. **Envio em tempo Real:** O Huffman Adaptativo é ótimo para enviar dados comprimidos em tempo real, se ajusta às frequências dos símbolos durante a transmissão e permite a reconstrução contínua no descompressor.
2. **Adaptabilidade** O Huffman Adaptativo funciona bem mesmo quando não sabemos o tamanho do arquivo antecipadamente. A Árvore de códigos é atualizada em tempo real, tornando os códigos associados os símbolos dinâmicos.

Descrição da Técnica

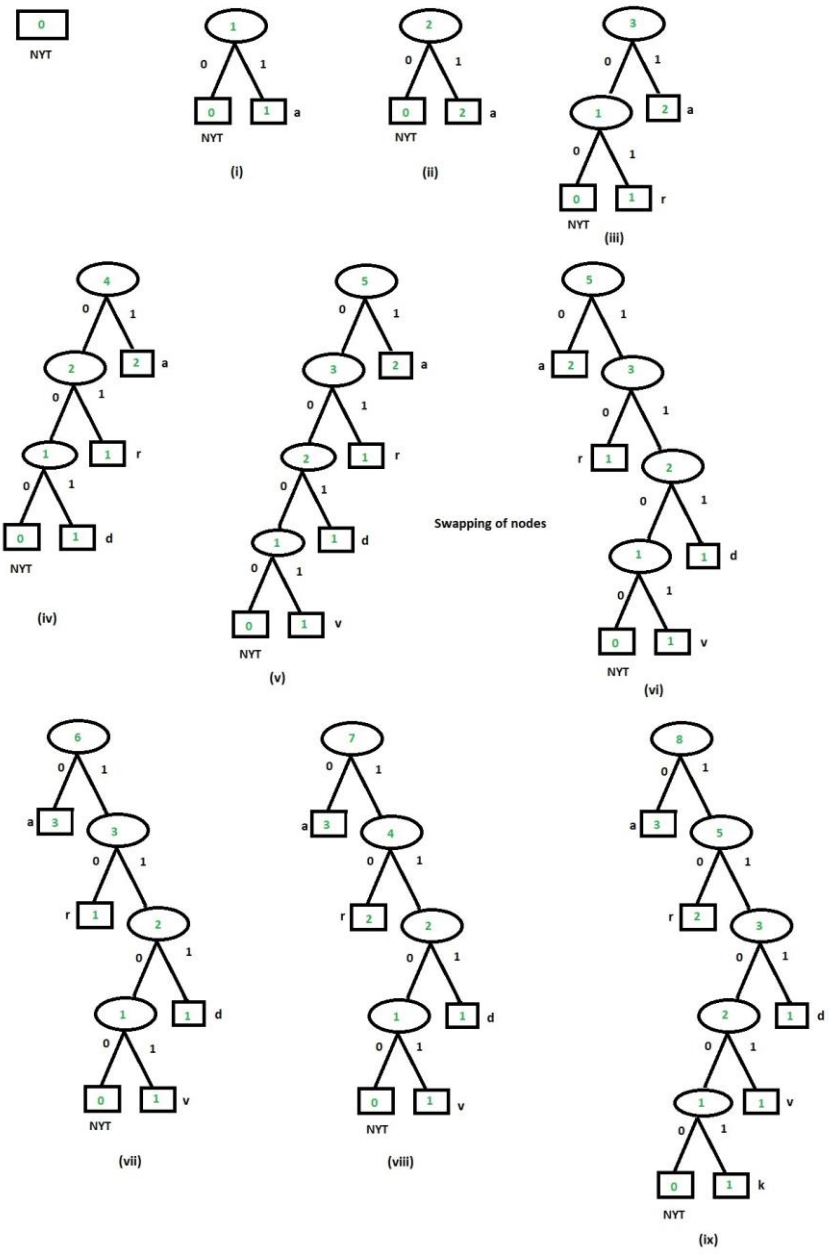
O compressor começa com uma árvore de Huffman vazia e quando um novo símbolo é inserido, cria-se um novo nó com o símbolo à direita e o símbolo especial à esquerda, o qual substitui o antigo símbolo especial. O primeiro símbolo de entrada é escrito no código de saída. O símbolo é então adicionado à árvore e um código é atribuído a ele. Na próxima vez em que esse símbolo for lido, seu código atual é escrito no código de saída e sua frequência é incrementada. A árvore é verificada e atualizada conforme necessário. No caso de um novo símbolo, o descompressor segue as mesmas etapas, utilizando a árvore para decodificar os códigos comprimidos.

Propriedade da Árvore de Huffman

A propriedade da árvore de Huffman é que, se a percorrermos nível por nível, verificando da esquerda para a direita e de baixo para cima, as frequências estarão em ordem crescente. O nó inferior esquerdo tem a menor frequência, e o superior direito tem a maior frequência. Essa propriedade garante que um símbolo com alta frequência de ocorrência deve ser atribuído a um código mais curto.

Exemplo

```
code = "aardvark"
The final Code we get is:
00000 1 010001 0000011 0001011 0 10  110001010
  a   a   r      d        v    a  r     k
```



Este exemplo foi retirado do site 'https://acervolima.com/codificacao-e-decodificacao-adaptativa-de-huffman/'.

Taxa de Compressão Comparando Implementações de Métodos

Definições das extensões usadas:

Para Textos:

- TXT: Um arquivo TXT é um formato que contém apenas texto simples, sem formatação.
- DOC: O arquivo DOC é usado pelo Microsoft Word para criar e editar documentos com texto, imagens e tabelas. Amplamente utilizado em documentos profissionais e acadêmicos.
- PDF: Formato de Documento Portátil. Um formato para representar documentos 2D de maneira independente de dispositivo e resolução. Ele pode funcionar como um contêiner para imagens comprimidas em JPEG, JPEG-2000, CCITT e outros formatos. Algumas versões do PDF se tornaram padrões ISO [Gonzalez & Woods, 2018].

Para Imagens:

- JPG: O padrão Joint Photographic Experts Group (JPEG) é usado para imagens de qualidade fotográfica. Seu sistema de codificação de linha de base com perda (mais comumente implementado) utiliza transformadas discretas de cosseno quantizadas (DCT) em blocos de imagem, codificação de Huffman e codificação de comprimento de execução. É um dos métodos mais populares para compressão de imagens na Internet [Gonzalez & Woods, 2018].
- BMP: Bitmap. Um formato de arquivo usado principalmente para imagens simples não comprimidas [Gonzalez & Woods, 2018].
- GIF: Graphic Interchange Format. Um formato de arquivo que utiliza a codificação LZW sem perda para imagens de 1 a 8 bits. É frequentemente usado para criar pequenas animações e filmes de baixa resolução para a Internet [Gonzalez & Woods, 2018].

Para Vídeos:

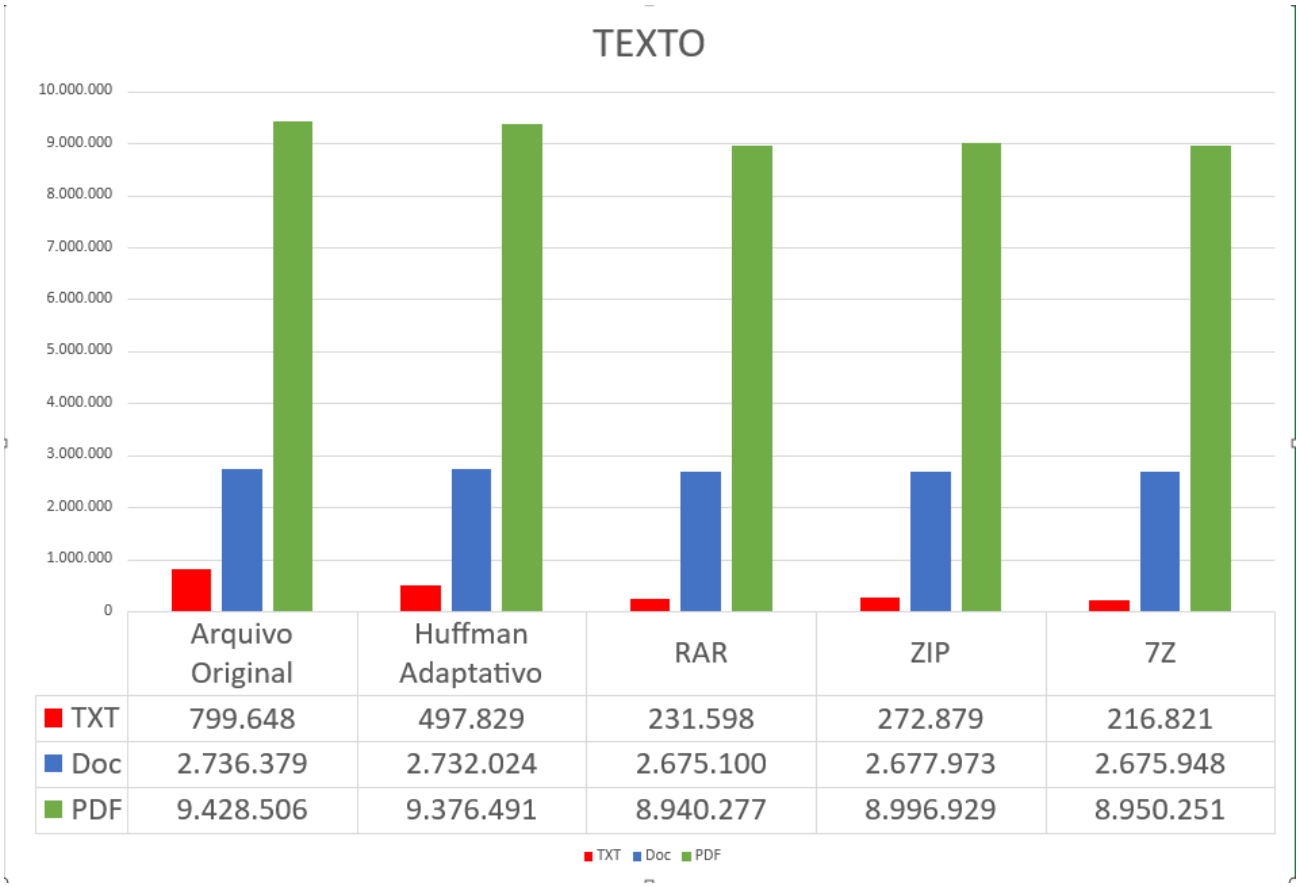
- MP4: Uma extensão do MPEG-2 que suporta tamanhos de bloco variáveis e diferenciação de predição [8.10] dentro dos quadros [Gonzalez & Woods, 2018].
- WEBM: O WebM é composto por vídeo codificado pelo codec VP8 e áudio codificado pelo codec Vorbis, ambos envolvidos no contêiner de arquivo Matroska. A motivação do Google é evitar pagar royalties - e fazer com que seus clientes paguem royalties - pelas licenças de propriedade intelectual (patentes) do MPEG-2 ou H.264. [Charles Poynton, 2013]
- AVI: Audio Video Interleave é um formato de arquivo de vídeo da Microsoft que comumente possui o tamanho maior que outras extensões devido a sua flexibilidade de suportar diversos codecs, dependendo da qualidade de áudio e vídeo.

Outros Métodos Encontrados

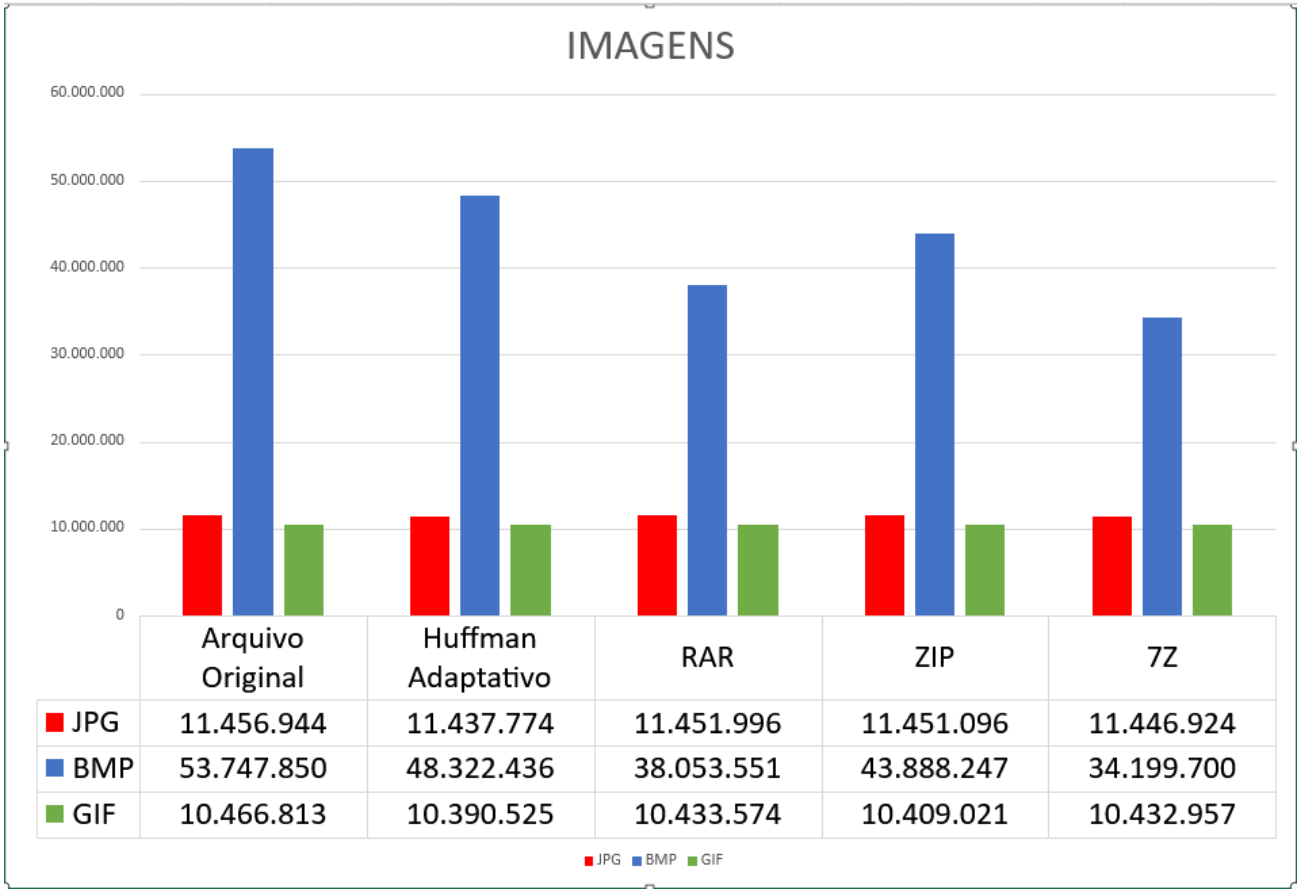
- RAR (Roshal Archive): É um formato de arquivo desenvolvido por Eugene Roshal, conhecido por sua alta taxa de compressão. Utiliza o método de compressão RAR compression algorithm e também pode utilizar algoritmos de codificação de árvores de Huffman para alcançar uma melhor eficiência de compactação.
- ZIP: É um formato de arquivo comumente utilizado para compactar e armazenar vários arquivos em um único arquivo. Ele utiliza o método de compressão DEFLATE, que é baseado em uma combinação de algoritmos LZ77 e Huffman, resultando em boa eficiência de compressão e velocidade de descompressão.
- 7Z (7-Zip): É um formato de arquivo de código aberto criado pelo programa de compressão 7-Zip. O 7Z utiliza o algoritmo LZMA2 (Lempel-Ziv-Markov chain-Algorithm), que é uma versão melhorada do algoritmo LZ77, oferecendo altas taxas de compressão e suporte para compressão sólida.

Resultados

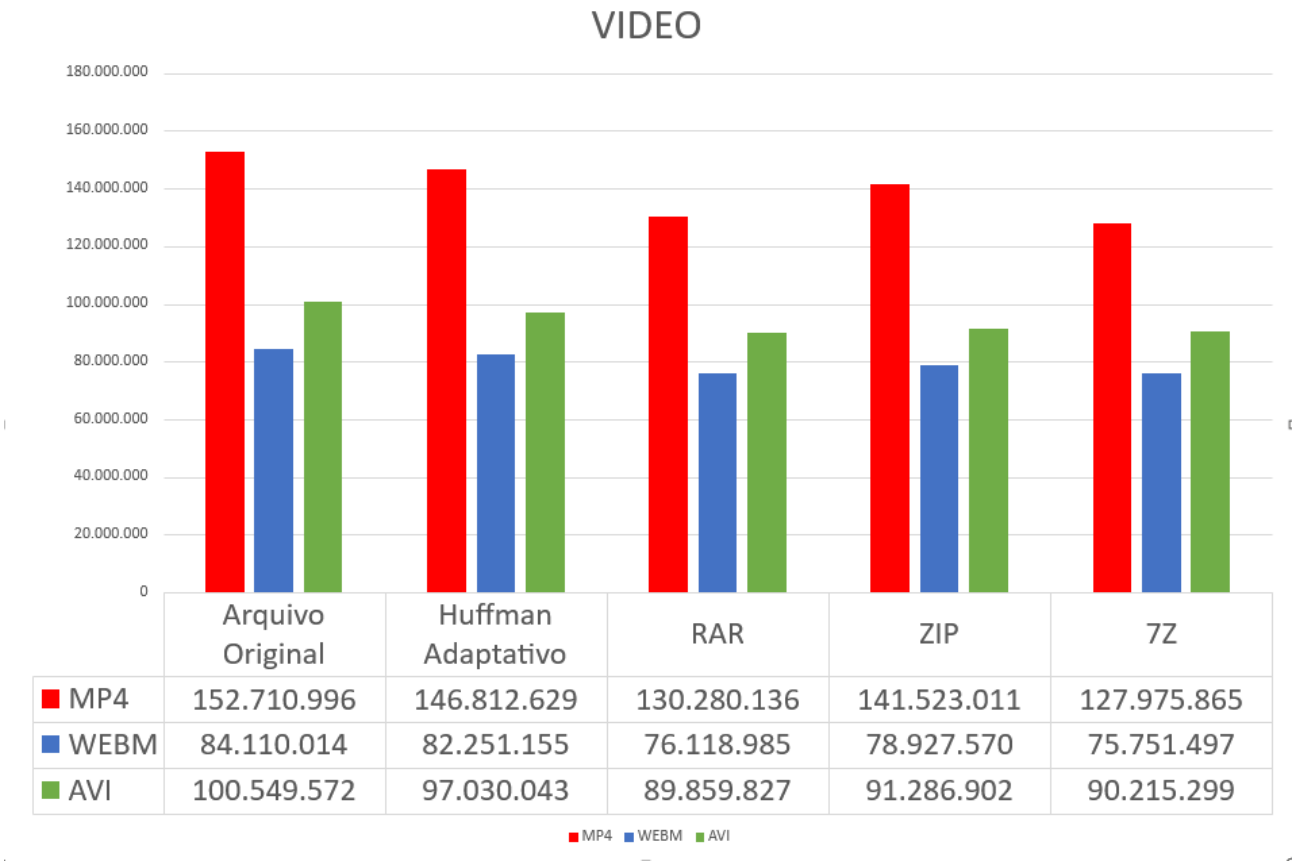
Tamanho dos arquivos em byte.



	Arquivo Original	Huffman Adaptativo	RAR	ZIP	7Z
TXT	799.648	497.829	231.598	272.879	216.821
Doc	2.736.379	2.732.024	2.675.100	2.677.973	2.675.948
PDF	9.428.506	9.376.491	8.940.277	8.996.929	8.950.251



	Arquivo Original	Huffman Adaptativo	RAR	ZIP	7Z
JPG	11.456.944	11.437.774	11.451.996	11.451.096	11.446.924
BMP	53.747.850	48.322.436	38.053.551	43.888.247	34.199.700
GIF	10.466.813	10.390.525	10.433.574	10.409.021	10.432.957



	Arquivo Original	Huffman Adaptativo	RAR	ZIP	7Z
MP4	152.710.996	146.812.629	130.280.136	141.523.011	127.975.865
WEBM	84.110.014	82.251.155	76.118.985	78.927.570	75.751.497
AVI	100.549.572	97.030.043	89.859.827	91.286.902	90.215.299

Análise de Compressão dos Algoritmos

Utilizando o algoritmo de Huffman dinâmico, obtivemos as seguintes taxas de compressão em relação aos arquivos de texto:

- O arquivo TXT ficou com 62,26% do tamanho original.
- O arquivo DOC ficou com 99,84% do tamanho original.
- O arquivo PDF ficou com 99,45% do tamanho original.

Já com o algoritmo RAR, as taxas de compressão foram as seguintes:

- O arquivo TXT ficou com 28,96% do tamanho original.
- O arquivo DOC ficou com 97,76% do tamanho original.
- O arquivo PDF ficou com 94,82% do tamanho original.

Com o algoritmo ZIP, as taxas de compressão foram:

- O arquivo TXT ficou com 34,12% do tamanho original.
- O arquivo DOC ficou com 97,87% do tamanho original.
- O arquivo PDF ficou com 95,42% do tamanho original.

Por fim, utilizando o algoritmo 7Z, obtivemos as seguintes taxas de compressão:

- O arquivo TXT ficou com 27,11% do tamanho original.
- O arquivo DOC ficou com 97,79% do tamanho original.
- O arquivo PDF terminou com 94,93% do tamanho original.

Observamos que o algoritmo de Huffman dinâmico apresentou as menores taxas de compressão para todos os arquivos.

Compressão para Arquivos de Imagens

Utilizando o algoritmo de Huffman dinâmico, obtivemos as seguintes taxas de compressão em relação aos arquivos JPG, BMP e GIF:

- O arquivo JPG ficou com 99,83% do tamanho original.
- O arquivo BMP ficou com 89,91% do tamanho original.
- O arquivo GIF ficou com 99,27% do tamanho original.

Com o algoritmo RAR, as taxas de compressão foram as seguintes:

- O arquivo JPG ficou com 99,96% do tamanho original.
- O arquivo BMP ficou com 70,8% do tamanho original.
- O arquivo GIF ficou com 99,68% do tamanho original.

Com o algoritmo ZIP, as taxas de compressão foram:

- O arquivo JPG ficou com 99,95% do tamanho original.
- O arquivo BMP ficou com 81,66% do tamanho original.
- O arquivo GIF ficou com 99,45% do tamanho original.

Por fim, utilizando o algoritmo 7Z, obtivemos as seguintes taxas de compressão:

- O arquivo JPG ficou com 99,91% do tamanho original.
- O arquivo BMP ficou com 63,63% do tamanho original.
- O arquivo GIF terminou com 99,68% do tamanho original.

Observamos que o algoritmo 7Z apresentou a maior taxa de compressão para os arquivos BMP, enquanto o algoritmo de Huffman dinâmico foi melhor para os outros.

Compressão para Arquivos de Vídeos

Utilizando o algoritmo de Huffman dinâmico, obtivemos as seguintes taxas de compressão em relação aos arquivos MP4, WebM e AVI:

- MP4: 96,14% do tamanho original.
- WebM: 97,79% do tamanho original.
- AVI: 96,5% do tamanho original.

Com o algoritmo RAR, as taxas de compressão foram as seguintes:

- MP4: 85,31% do tamanho original.
- WebM: 90,5% do tamanho original.
- AVI: 89,37% do tamanho original.

Com o algoritmo ZIP, as taxas de compressão foram:

- MP4: 92,67% do tamanho original.
- WebM: 93,84% do tamanho original.
- AVI: 90,79% do tamanho original.

Por fim, utilizando o algoritmo 7Z, obtivemos as seguintes taxas de compressão:

- MP4: 83,8% do tamanho original.
- WebM: 90,06% do tamanho original.
- AVI: 89,72% do tamanho original.

Observamos que o algoritmo de Huffman dinâmico teve os piores resultados para os arquivos de vídeo.

Conclusão

O algoritmo de Huffman adaptativo teve a pior compressão em comparação com os outros algoritmos usados. Isso acontece porque ele é mais simples e menos eficiente do que RAR, ZIP e 7Z, que são métodos mais avançados de compressão.

Apesar de ter as piores taxas de compressão, o Huffman adaptativo ainda foi eficiente para arquivos JPG e GIF, que possuem redundância e padrões que o algoritmo pode explorar. Mesmo assim, em geral, ele não é tão eficiente quanto 7Z,RAR e ZIP, especialmente para diferentes tipos de arquivos. A escolha do algoritmo adequado depende do tipo de arquivo e dos resultados desejados em termos de compressão.

Referências:

Poynton, C. (2013). *Digital Video and HD: Algorithms and Interfaces*. Morgan Kaufmann.

Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing*. Pearson.

7-Zip. (2023). Em Wikipedia. Recuperado de https://pt.wikipedia.org/wiki/7-Zip

MacKay, D. (2003). *Information Theory, Inference, and Learning Algorithms*.

Codificação e decodificação adaptativa de Huffman. (2023). Recuperado de https://acervolima.com/codificacao-e-decodificacao-adaptativa-de-huffman/

Codificação de Huffman e compressão de dados. (2023). Recuperado de https://bitismyth.wordpress.com/2016/01/22/codificacao-de-huffman-e-compressao-de-dados/