

UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE CIÊNCIAS - CAMPUS BAURU
DEPARTAMENTO DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

GUSTAVO YUJII SILVA KADOOKA

**EXPLORANDO ALGORITMOS DE COMPRESSÃO DE DADOS:
TEORIA, IMPLEMENTAÇÃO E DESEMPENHO**

BAURU
Novembro/2025

GUSTAVO YUJII SILVA KADOOKA

**EXPLORANDO ALGORITMOS DE COMPRESSÃO DE DADOS:
TEORIA, IMPLEMENTAÇÃO E DESEMPENHO**

Trabalho de Conclusão de Curso em
Ciência da Computação apresentado
ao Departamento de Computação da
Universidade Estadual Paulista “Júlio de
Mesquita Filho”, Faculdade de Ciências,
Campus Bauru.

Orientador: Profa. Dra. Andréa Carla
Gonçalves Vianna

K11e

Kadooka, Gustavo Yujii Silva

Explorando Algoritmos de Compressão de Dados: Teoria,
Implementação e Desempenho / Gustavo Yujii Silva Kadooka. --
Bauru, 2025

60 p. : il., tabs.

Trabalho de conclusão de curso (Bacharelado - Ciência da
Computação) - Universidade Estadual Paulista (UNESP), Faculdade
de Ciências, Bauru

Orientadora: Andréa Carla Gonçalves Vianna

1. Compressão de dados. 2. Teoria da informação. 3. Compressão
sem perdas – Huffman, LZ77, LZW, GZIP. I. Título.

Gustavo Yujii Silva Kadooka

Explorando Algoritmos de Compressão de Dados: Teoria, Implementação e Desempenho

Trabalho de Conclusão de Curso em Ciência da Computação apresentado ao Departamento de Computação da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Faculdade de Ciências, Campus Bauru.

Banca Examinadora

**Profa. Dra. Andréa Carla Gonçalves
Vianna**

Orientador

Universidade Estadual Paulista “Júlio de
Mesquita Filho”

Faculdade de Ciências

Departamento de Ciência da Computação

Professor Convidado 1

Universidade Estadual Paulista “Júlio de
Mesquita Filho”

Faculdade de Ciências

Departamento de Ciência da Computação

Professor Convidado 2

Universidade Estadual Paulista “Júlio de
Mesquita Filho”

Faculdade de Ciências

Departamento de Ciência da Computação

Bauru, 11 de novembro de 2025.

À minha amada família, fonte inesgotável de inspiração e força.

Agradecimentos

Agradeço à Universidade Estadual Paulista “Júlio de Mesquita Filho” – Faculdade de Ciências, Campus de Bauru –, por ter sido o espaço onde pude me desenvolver como estudante, profissional e indivíduo. Sou grato a todos os professores e funcionários da instituição pelo conhecimento compartilhado e pelo apoio ao longo da graduação.

À minha orientadora, Profa. Dra. Andréa Carla Gonçalves Vianna, pela disponibilidade, paciência diante das dificuldades e por acreditar no meu potencial durante o desenvolvimento deste trabalho.

Aos meus colegas de curso, que estiveram presentes ao longo desta jornada, dividindo aprendizados, desafios e conquistas. O convívio e a troca de experiências tornaram a caminhada mais leve e significativa.

À minha irmã Gabriely, pelas palavras de incentivo e carinho em todos os momentos, mesmo nos mais difíceis.

E, com o meu mais profundo e sincero agradecimento, menciono especialmente meus pais – Flávia e Silvio –, por todo o apoio emocional, pelos ensinamentos de vida, pela confiança depositada em mim e por sempre me encorajarem a seguir em frente. Esta conquista é, acima de tudo, de vocês.

Resumo

Este trabalho apresenta um estudo abrangente, a implementação prática e a análise de desempenho de algoritmos clássicos de compressão de dados, a saber: Huffman, LZ77, LZW e GZIP. São explorados os conceitos teóricos fundamentais de cada algoritmo, seguidos pela implementação prática utilizando a linguagem de programação C++. As implementações dos algoritmos lidam com os formatos de arquivo de texto simples (.txt), imagens bitmap (.bmp) e arquivos de áudio PCM (.wav), demonstrando versatilidade e aplicabilidade em diferentes domínios de dados. Para facilitar a interação do usuário, foi desenvolvida uma interface gráfica utilizando a biblioteca GTK, permitindo a seleção dos algoritmos de compressão e o arquivo de entrada. Ademais, para realizar uma análise comparativa rigorosa, foram utilizados scripts em Python para processar os dados experimentais e gerar representações gráficas dos principais indicadores de desempenho, como a taxa de compressão e o tempo de execução para diferentes tipos de arquivos. Os resultados experimentais indicam que não há um algoritmo que se destaque universalmente, pois cada um apresenta vantagens específicas conforme o tipo de arquivo e as características dos dados. Isso ressalta a importância da escolha do método de compressão adequado às necessidades específicas da aplicação. O projeto contribui tanto com uma ferramenta prática para compressão de dados quanto com um recurso educacional que auxilia na compreensão dos algoritmos clássicos e seu comportamento em cenários reais.

Palavras-chave: Compressão de dados. Algoritmos de compressão. Huffman. LZ77. LZW. GZIP. C++. Python. GTK.

Abstract

This work presents a comprehensive study, implementation, and performance analysis of classical data compression algorithms, namely Huffman, LZ77, LZW, and GZIP. The theoretical concepts behind each algorithm are explored in depth, followed by practical implementations using C++ programming language. The implementations support plain text (.txt), bitmap images (.bmp), and PCM audio files (.wav), demonstrating versatility and applicability in different data domains. To facilitate user interaction, a graphical user interface (GUI) was developed with the GTK library, enabling selection of compression algorithms, and input files. Furthermore, to conduct a rigorous comparative analysis, Python scripts were employed to process experimental data and generate graphical representations of key performance metrics such as compression ratio and execution time for different file types. The experimental results indicate that no single algorithm universally outperforms others; rather, each one exhibits specific advantages depending on the file type and data characteristics. This highlights the importance of choosing the appropriate compression method tailored to the application's needs. The project contributes as a practical tool for data compression and an educational resource that aids in understanding fundamental compression algorithms and their behavior in real-world scenarios.

Keywords: Data compression, compression algorithms, Huffman coding, LZ77, LZW, GZIP, C++, Python, GTK.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

LZ77	Lempel-Ziv 1977 (algoritmo de compressão)
LZW	Lempel-Ziv-Welch (algoritmo de compressão)
GZIP	GNU zip (algoritmo de compressão)
GTK	GIMP Toolkit (biblioteca gráfica para interface)

Sumário

1	INTRODUÇÃO	12
1.1	Problema	13
1.2	Objetivos	13
1.2.1	Objetivo geral	13
1.2.2	Objetivos específicos	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Teoria da informação	15
2.2	Compressão de dados	15
2.3	Algoritmos clássicos de compressão sem perdas	15
3	IMPLEMENTAÇÃO	16
4	ANÁLISE COMPARATIVA	17
5	CONCLUSÃO	18
	REFERÊNCIAS	19

1 Introdução

A compressão de dados é uma técnica essencial na ciência da computação, utilizada para reduzir o tamanho dos arquivos, otimizando o uso de recursos e acelerando a transmissão de dados – aspectos cruciais na era digital em que vivemos.

O desenvolvimento da área teve início na década de 1950, com os primeiros métodos voltados à otimização do espaço de armazenamento. Entre estes métodos, destaca-se o algoritmo de Huffman (HUFFMAN, 1952), que emprega uma codificação de prefixo variável, atribuindo códigos menores aos símbolos mais frequentes e maiores aos menos frequentes. Essa abordagem se consolidou como um dos pilares da área e é utilizada em formatos como ZIP e GZIP.

Posteriormente, novos algoritmos surgiram. Um dos mais significativos é o LZ77, proposto por Abraham Lempel e Jacob Ziv (ZIV; LEMPEL, 1977), que adota uma estratégia baseada em dicionários, substituindo sequências repetidas por referências a posições anteriores. A partir desta ideia, o LZW vem à tona aprimorando o LZ77 ao construir dinamicamente um dicionário de *strings* durante a compressão (WELCH, 1984), sendo amplamente utilizado em formatos como GIF e TIFF.

Outro algoritmo relevante é o GZIP, amplamente adotado em sistemas Unix e aplicações web. Ele combina a eficiência da codificação de Huffman com o LZ77, alcançando boas taxas de compressão sem perdas. Sua popularidade decorre da combinação de alta eficácia com a facilidade de descompressão.

Apesar do surgimento de novas técnicas, os algoritmos de Huffman, LZ77, LZW e GZIP continuam a formar a espinha dorsal dos métodos clássicos de compressão de dados e são amplamente utilizados em diversas aplicações.

Nos últimos anos, surgiram algoritmos mais modernos, como o Brotli – desenvolvido pela Google – que combina codificação de Huffman e transformações de fluxo de dados para obter taxas de compressão superiores às do GZIP (ALAKUIJALA et al., 2016). Outro exemplo é o Zstandard (Zstd), criado pelo Facebook, que alia velocidade e compressão eficiente, sendo adequado tanto para compressão em tempo real quanto para grandes volumes de dados (COLLET, 2016).

Além disto, algoritmos baseados em aprendizado de máquina têm ganhado destaque, principalmente na compressão de imagens e áudio. Redes neurais como GANs (para imagens) e modelos como o WaveNet (para áudio) (DeepMind, 2016) vêm sendo explorados para melhorar a qualidade das compressões. No campo da compressão de vídeo, codecs como HEVC (H.265) e VVC vêm sendo desenvolvidos

para oferecer compressões mais eficazes em vídeos de alta definição.

A compressão de dados desempenha, portanto, um papel estratégico em diversas áreas tecnológicas, impactando diretamente sistemas de armazenamento, transmissão de dados, streaming, redes e computação em nuvem. A análise de desempenho dos algoritmos, considerando métricas como taxa de compressão e tempo de execução, é essencial para a escolha do método mais adequado a cada situação.

Neste contexto, apresenta-se um estudo aprofundado sobre os algoritmos clássicos de compressão Huffman, LZ77, LZW e GZIP. São abordados os fundamentos teóricos de cada método, suas implementações práticas em linguagem C++ com suporte a arquivos nos formatos .txt, .bmp e .wav, e uma interface gráfica desenvolvida com GTK. A avaliação experimental é realizada por meio de comparações de desempenho, com auxílio de scripts em Python para geração de gráficos e análise dos resultados. O objetivo é compreender o comportamento dos algoritmos em diferentes tipos de dados, destacando suas vantagens, limitações e aplicações mais adequadas.

O restante deste trabalho está organizado da seguinte forma: o Capítulo 2 apresenta os fundamentos teóricos sobre compressão de dados, abordando a Teoria da Informação e os algoritmos Huffman, LZ77, LZW e GZIP. O Capítulo 3 descreve a implementação dos algoritmos, incluindo aspectos técnicos e ferramentas utilizadas. No Capítulo 4, é realizada uma análise comparativa com base nos testes realizados, considerando taxa de compressão e tempo de execução. Por fim, o Capítulo 5 apresenta as conclusões do trabalho e sugestões para pesquisas futuras.

1.1 Problema

Como os diferentes algoritmos clássicos de compressão de dados (Huffman, LZ77, LZW e GZIP) se comparam em termos de eficiência de compressão e tempo de execução, e qual o impacto dessas métricas em aplicações práticas que lidam com texto, imagem e áudio?

1.2 Objetivos

1.2.1 Objetivo geral

Investigar e comparar o desempenho de algoritmos clássicos de compressão de dados sem perdas, a saber: Huffman, LZ77, LZW e GZIP, por meio da implementação prática dessas técnicas, aplicando-as a diferentes tipos de arquivos e analisando sua eficiência em termos da taxa de compressão e tempo de execução.

1.2.2 Objetivos específicos

- Estudar os fundamentos teóricos dos algoritmos de compressão sem perdas selecionados, compreendendo o seu funcionamento, estrutura e aplicação histórica.
- Implementar, em linguagem C++, os algoritmos Huffman, LZ77, LZW e GZIP, com foco na compressão e descompressão de arquivos de texto (.txt), imagem bitmap (.bmp) e áudio PCM (.wav).
- Desenvolver uma interface gráfica com a biblioteca GTK, que permita ao usuário escolher o algoritmo e o algoritmo de entrada de forma interativa.
- Criar scripts em Python para a análise comparativa, organizando e visualizando os resultados experimentais por meio de gráficos e tabelas.
- Avaliar os pontos fortes e fracos de cada algoritmo, considerando o tipo de dado, a estrutura do arquivo, o tempo de processamento e a taxa de compressão.
- Identificar o algoritmo mais adequado a cada cenário específico, com base nos resultados experimentais obtidos.

2 Fundamentação teórica

Este capítulo apresenta os fundamentos teóricos que embasam o desenvolvimento deste trabalho. Para compreender o funcionamento e as diferenças entre os algoritmos de compressão de dados abordados, é necessário introduzir alguns conceitos essenciais.

Primeiramente, será apresentada a teoria da informação, a qual fornece o embasamento matemático para a compressão de dados sem perdas, por meio de conceitos como entropia e redundância. Em seguida, serão discutidos os princípios gerais da compressão de dados, com foco nas técnicas sem perdas, suas aplicações e métricas de desempenho. Por fim, serão detalhados os algoritmos clássicos de compressão sem perdas utilizados neste trabalho: Huffman, LZ77, LZW e GZIP, incluindo suas características, funcionamento e aplicações práticas.

A estrutura deste capítulo está organizada da seguinte forma:

- **Seção 3.1** – Apresenta os principais conceitos da Teoria da Informação, com ênfase na entropia de Shannon e seu papel na compressão de dados;
- **Seção 3.2** – Discute os fundamentos da compressão de dados sem perdas, destacando seus objetivos, métricas e distinções em relação à compressão com perdas;
- **Seção 3.3** – Descreve detalhadamente os algoritmos Huffman, LZ77, LZW e GZIP, explicando seus princípios de funcionamento, vantagens, limitações e casos de uso.

2.1 Teoria da informação

2.2 Compressão de dados

2.3 Algoritmos clássicos de compressão sem perdas

3 Implementação

4 Análise comparativa

5 Conclusão

Referências

ALAKUIJALA, J.; SZABADKA, Z.; VANDEVENNE, L.; FARRUGGIA, R.; KLIUCHNIKOV, E.; SZABADKAI, I.; ASSCHE, Z. V. Brotli compressed data format. In: INTERNATIONAL WORLD WIDE WEB CONFERENCES STEERING COMMITTEE. *Proceedings of the 25th International Conference on World Wide Web*. [S.l.], 2016. p. 1593–1603.

COLLET, Y. *Zstandard Compression Algorithm*. 2016. Disponível em: <<https://facebook.github.io/zstd/>>.

DeepMind. *WaveNet: A generative model for raw audio*. 2016. Disponível em: <<https://deepmind.google/research/breakthroughs/wavenet/>>.

HUFFMAN, D. A. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, Institute of Radio Engineers, v. 40, n. 9, p. 1098–1101, 1952.

WELCH, T. A. A technique for high-performance data compression. *Computer*, IEEE, v. 17, n. 6, p. 8–19, 1984.

ZIV, J.; LEMPEL, A. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, IEEE, v. 23, n. 3, p. 337–343, 1977.