

Estudo e Implementação dos Algoritmos de Compressão LZ77 e Codificação Aritmética na Biblioteca Komm

RESUMO EXPANDIDO - Disciplina de TCC029009

Rhenzo Hideki Silva Kajikawa

Estudante do Curso de Engenharia de Telecomunicações

Roberto Wanderley da Nobrega, Dr.

Professor orientador

Semestre 2025.1

Resumo- *A biblioteca de compressão de dados Komm será estendida com a integração de dois algoritmos sem perdas: LZ77 e codificação aritmética. O cenário de redes de telecomunicações, marcado pelo aumento exponencial de dados, exige técnicas eficientes de compressão para otimização de largura de banda e armazenamento. Os objetivos englobam a estudo teórica dos algoritmos, projeto e implementação de módulos na arquitetura existente em Python, documentação e validação por meio de testes automatizados de correto funcionamento e desempenho. A metodologia inclui revisão bibliográfica, desenvolvimento de software e análise comparativa, visando quantificar ganhos em taxa de compressão, tempo de execução e uso de memória. Espera-se comprovar a viabilidade prática dos algoritmos e oferecer subsídios para futuras ampliações da biblioteca.*

Palavras-chave: Compressão sem perdas; LZ77; codificação aritmética; Python.

1 Introdução

A compressão de dados é essencial para minimizar custos de armazenamento e transmissão, reduzindo o volume de dados sem comprometer o entendimento da informação. No caso da compressão sem perdas, segundo a teoria de Shannon, a entropia da fonte estabelece o limite inferior para a taxa média de bits de qualquer esquema de compressão (MACKAY, 2003). Para aproximar-se desse limite teórico, algoritmos de compressão combinam técnicas de dicionário e codificação estatísticas.

Na prática, formatos de compressão amplamente utilizados empregam essa abordagem híbrida. Por exemplo, o algoritmo DEFLATE, utilizado no formato ZIP, aplica LZ77 em conjunto com a codificação de Huffman para obter compressões melhores. Essa combinação explora tanto os padrões repetitivos de longo alcance quanto as probabilidades de ocorrência dos símbolos, elevando a eficiência global do método (DEUTSCH, 1996).

De modo geral, os algoritmos de compressão sem perdas dividem-se em métodos de codificação estatística e métodos de dicionário (SAYOOD, 2012). Por exemplo, a codificação de Huffman e a codificação aritmética são técnicas estatísticas – esta última introduzida por Witten, Neal e Cleary (1987), capaz de representar toda a mensagem como um único número fracionário no intervalo $[0,1)$, alcançando compressões muito próximas do limite de entropia. Por sua vez, os métodos de dicionário exploram redundâncias substituindo sequências repetitivas por referências a ocorrências anteriores já vistas: um dos mais conhecidos é o algoritmo LZ77, proposto por Ziv e Lempel (1977), que implementa esse princípio construindo dinamicamente um “dicionário” de padrões enquanto lê os dados.

Apesar do uso disseminado dessas técnicas em aplicações, a biblioteca de código aberto Komm não dispunha até o momento de uma implementação do LZ77 e nem de codificação aritmética. Essa biblioteca voltada ao ensino e à simulação em comunicação digital, já inclui diversos algoritmos clássicos de compressão, como os código de Huffman, Shannon–Fano e LZ78, evidenciando a relevância de incorporar as técnicas LZ77 e codificação aritmética para torná-la mais completa. Portanto, este trabalho tem como objetivo desenvolver e integrar um versão do LZ77 e um codificador aritmético na biblioteca Komm, avaliando estes algoritmos em termos de taxa de compressão, tempo de processamento e uso de memória.

1.1 Objetivo geral

Expandir as capacidades da biblioteca Komm com a implementação dos algoritmos LZ77 e codificação aritmética.

1.2 Objetivos específicos

- Estudo da literatura sobre o algoritmo LZ77 e suas variantes, identificando aspectos de implementação e aplicações.
- Estudo do algoritmo de codificação aritmética, contemplando modelagem probabilística, renormalização e limitações de precisão.
- Projetar e desenvolver módulos LZ77 e aritmético na arquitetura da Komm, observando padrões de codificação e cobertura de testes e escrita da documentação.

- Validar a compressão e descompressão em arquivos de textos e imagens, assegurando corretude e integridade.
- Realizar análise comparativa de desempenho (taxa de compressão, tempo de execução e uso de memória) em relação a Huffman, LZ78 e LZW.

2 Metodologia

A metodologia adotada iniciará com uma ampla revisão bibliográfica sobre os algoritmos LZ77 e aritmético, envolvendo a análise de trabalhos clássicos e recentes que descrevem princípios, variantes e otimizações de cada técnica. Em seguida, procedeu-se ao projeto de software, definindo as APIs e estruturas de dados em Python compatíveis com a arquitetura modular da biblioteca Komm. Esse projeto incluiu a elaboração de testes unitários que garantirão a correção de cada componente. Na fase de implementação, os módulos de compressão serão desenvolvidos e integrados ao repositório principal, observando padrões de codificação e boas práticas de documentação inline. A etapa de validação consistirá na execução de testes de compressão e descompressão em conjuntos de arquivos reais — como textos, logs e imagens simples —, com mensuração de métricas de desempenho (taxa de compressão, tempo de processamento e uso de memória) e verificação da integridade dos dados reconstruídos. Por fim, uma análise comparativa sistemática, confrontando os resultados obtidos por LZ77 e codificação aritmética com os dos algoritmos Huffman, LZ78 e LZW já presentes na biblioteca, de modo a quantificar ganhos relativos e identificar cenários de maior eficácia para cada método.

3 Resultados esperados

São esperados os seguintes resultados ao término do trabalho: a implementação funcional dos algoritmos LZ77 e de codificação aritmética; a avaliação empírica de suas taxas de compressão e tempos de processamento em diferentes tipos de dados; e a análise comparativa do desempenho de cada algoritmo, confrontando os resultados obtidos com o limite teórico de compressão (entropia) e com métodos tradicionais de compressão sem perdas. Esses resultados permitirão verificar em que medida os algoritmos desenvolvidos se aproximam da eficiência teórica prevista e das ferramentas de compressão existentes, contribuindo para a compreensão de suas vantagens e limitações em diferentes cenários de uso.

Referências

DEUTSCH, L. P. *DEFLATE Compressed Data Format Specification version 1.3*. RFC Editor, 1996. RFC 1951. (Request for Comments, 1951). Disponível em: <<https://www.rfc-editor.org/info/rfc1951>>.

MACKAY, D. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003. ISBN 9780521642989. Disponível em: <<https://books.google.com.br/books?id=AKuMj4PN EMC>>.

SAYOOD, K. *Introduction to Data Compression*. Elsevier Science, 2012. (The Morgan Kaufmann Series in Multimedia Information and Systems). ISBN 9780124157965. Disponível em: <<https://books.google.com.br/books?id=mkCMxnHm6hsC>>.