Proceeding Series of the Brazilian Society of Applied and Computational Mathematics, Vol. 3, N. 1, 2015.
Trabalho apresentado no XXXV CNMAC, Natal-RN, 2014.

Compressão de textos

Fabrício Aranda Negri **Marco Antônio Piteri** **Aylton Pagamisse**
Departamento de Matemática e Computação, FCT, UNESP,
19060-900, Presidente Prudente, SP.
E-mail: fabricioaranda@gmail.com, piteri@fct.unesp.br, aylton@fct.unesp.br

RESUMO

O problema da compressão de dados nos remete ao ano de 1838, quando Samuel Morse inventou um código para representar letras, números e sinais de pontuação para o sistema de telégrafos. Esta codificação pode ser considerada como uma das primeiras tentativas de utilizar código no intuito de representar um conjunto de caracteres em um formato menor. Entretanto, apenas em 1948 foi que Claude Elwood Shannon estabeleceu um estudo matemático sobre a comunicação e suas propriedades [3], considerado o marco inicial para o desenvolvimento da Teoria da Informação, em particular da compressão de dados. Esta última tem como propósito reduzir o espaço ocupado por dados em dispositivos de armazenamento, com o consequente ganho de desempenho (através da redução do tempo) em transmissões.

A compressão de dados existe desde os primórdios da Ciência da Computação, onde recursos de memória eram escassos e extremamente caros, assim, qualquer possibilidade de minimizar o espaço ocupado em disco era bem-vinda. Uma alternativa imediata era simplesmente tentar codificar o conteúdo do arquivo a ser armazenado. Se o resultado dessa codificação fosse menor que o arquivo original, haveria uma economia de memória. Obviamente a tabela de códigos deveria ser armazenada para permitir o inverso da codificação do arquivo, referenciado na literatura por decodificação.

Essencialmente, maiores taxas de transmissão de dados entre dois pontos quaisquer podem ser obtidas de duas maneiras: aumentando-se fisicamente a largura de banda (canal) por onde os dados são transferidos ou por meio de alguma manipulação nos próprios dados, reduzindo-se de algum modo à quantidade deles e consequentemente, o tempo despendido. Técnicas de compressão de dados são usadas ainda no armazenamento de dados (texto, som, imagem) em DVDs (Digital Video Disks) e também estão presentes na tecnologia de televisão digital, onde grandes quantidades de dados (vídeo sobre demanda) estão envolvidos. Neste tipo de aplicação, a velocidade de descompressão deve ser feita em tempo real, exigindo algoritmos de decodificação extremamente eficientes. Em suma, a compressão de dados é, no contexto da Ciência da Computação, a ciência (e arte) de representar a informação em uma forma compacta e tem sido uma das tecnologias que permitiu a revolução digital multimídia ao longo do tempo [2].

Este trabalho tem como objetivo o desenvolvimento de um sistema de software que implementa quatro técnicas clássicas de compressão de textos (Run-length, Huffman [1], LZSS [4] e LZW). Experimentos comparativos entre os métodos abordados e alguns dos softwares de compressão mais conhecidos no mercado (Winzip, WinRAR e 7-Zip) foram realizados para aferir a qualidade do sistema proposto.

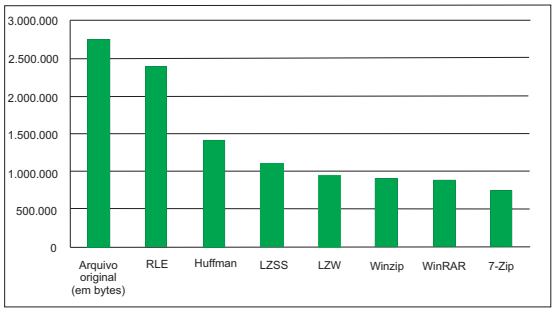
Os materiais utilizados neste trabalho foram a ferramenta de programação Builder C++ (para o desenvolvimento do software) e um notebook de uso pessoal. A realização dos experimentos pressupõe o uso de arquivos de texto em diferentes alfabetos, incluindo aqueles com genomas disponíveis no site do *European Bioinformatics Institute* (http://www.ebi.ac.uk). Arquivos da base de dados *Canterbury corpus* (http://corpus.canterbury.ac.nz) e do projeto Gutenberg (http://www.gutenberg.org) também foram utilizados.

A partir das Figuras 1 e 2, é possível depreender que alguns dos métodos de compressão implementados neste trabalho obtiveram resultados próximos aos dos softwares comerciais (que fazem uso de mais de uma técnica de compressão). Pode-se notar ainda que os gráficos da taxa média de compressão

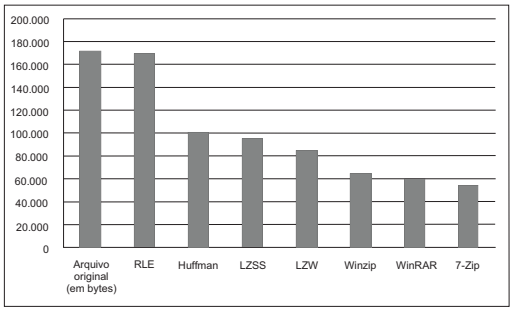
DOI: 10.5540/03.2015.003.01.0361

das diferentes bases de dados possuem, de um modo geral, comportamentos semelhantes. O tamanho do alfabeto pode influenciar em alguns métodos, como por exemplo no Run-length. Se compararmos o desempenho deste método nas bases EBI e *Canterbury corpus* (Figuras 1(a) e 1(b), respectivamente), observamos que em alfabetos menores (base EBI) a técnica Run-length alcança melhores resultados.

As Figuras 1(a) e 1(b) ilustram as taxas médias de compressão obtidas nos experimentos com uma coleção de arquivos das bases EBI e *Canterbury corpus*.



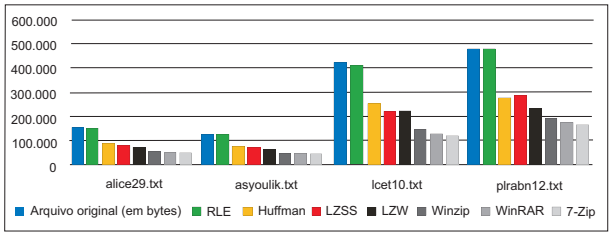
(a) *European Bioinformatics Institute*



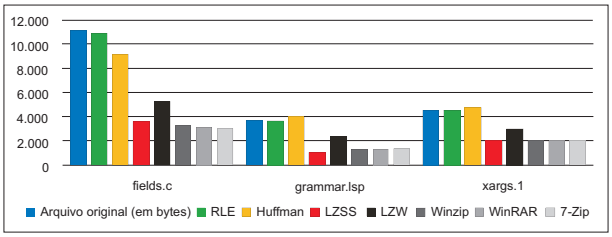
(b) Base *Canterbury corpus*

Figura 1: Taxa média da compressão de arquivos

As Figuras 2(a) e 2(b) apresentam as taxas de compressão obtidas por cada método em cada um dos arquivos da base *Canterbury corpus*.



(a)



(b)

Figura 2: Compressão dos arquivos da base *Canterbury corpus*

Em linhas gerais, a partir dos experimentos realizados conclui-se que os resultados obtidos foram próximos aos dos principais softwares comerciais de compressão disponíveis no mercado e estão em consonância com os objetivos inicialmente propostos para essa fase inicial em que se encontra o projeto. Em alguns casos, a utilização de apenas um método é suficiente para atingir níveis satisfatórios de compressão, desde que o conjunto de dados possua características favoráveis, tais como: o tamanho do alfabeto, disposição dos caracteres dentro do arquivo a ser compresso, tamanho do arquivo, entre outros.

Palavras-chave: *Teoria da Informação, Codificação de dados, Compressão de dados*

Referências

[1] D.A. Huffman, A method for the construction of minimum-redundancy codes, em “Proceedings of the I.R.E.” pp. 1098-1101, 1951.

[2] K. Sayood, “Introduction to Data Compression”, San Francisco, CA, 2006.

[3] C. E. Shannon, A mathematical theory of communication, *Bell System Technical Journal*, 27 (1948) 379-423, 623-656.

[4] J. A. Storer,; T.G. Syzmanski, Data compression via textual substitution, *Journal of the ACM*, 29 (1982) 928-951.

