

# Compressão de dados utilizando codificação de Huffman

Colocar o seu nome

Departamento de Sistemas e Computação  
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brazil

seunome@inf.furb.br

**Abstract.** *This article proposes a solution to the problem of data compression on file. The solution was developed through the implementation of a simple algorithm that performs data compression based on the coding Huffman.*

**Resumo.** *Este artigo propõe uma solução para o problema de compressão de dados em arquivo. A solução foi desenvolvida através da implementação de um algoritmo simples que realiza a compressão dos dados tendo como base a codificação de Huffman.*

## 1. Introdução

Nos últimos anos, com o avanço da tecnologia, aumentam cada vez mais os problemas de armazenamento e transferência de informações. Desde os primórdios da internet, com a transmissão de arquivos pela rede, viu-se a necessidade de diminuir o máximo possível as informações, evitando o envio de informação desnecessária, e otimizando a transmissão e por consequência diminuindo o tráfego de rede.

Muitas alternativas foram propostas para solucionar o problema de transmissão de arquivos. A compressão das informações contidas no arquivo é uma delas. Segundo ARRUDA e GOES (2003) a compressão pode ser lógica ou física, com ou sem perda de dados.

Este trabalho propõe uma solução para compressão de dados utilizando a codificação de Huffman que se trata de um método de compactação que usa as probabilidades de ocorrência dos símbolos no conjunto de dados a ser compactado para determinar códigos de tamanho variável para cada símbolo.

## 2. Desenvolvimento

O desenvolvimento do algoritmo se deu em duas etapas. Na primeira etapa, o arquivo é aberto e copiado todo seu conteúdo. A partir desta informação, cada caractere contido no arquivo é transformado em sua representação binária. Caso o caractere seja um dos caracteres que serão utilizados na compressão, a representação binária passa a ser o valor previamente definido, conforme Tabela 1.

Representação decimal / binário		
Caracter	Decimal	Binário
a	97	1
b	98	000
c	99	01

Tabela 1: Representação binária

Desta forma, o valor do caractere “a” que originalmente é representado em binário por 1100001, será representado por 1. Após todos os caracteres presente no arquivo serem percorridos, e quando possível otimizado, termina a primeira etapa do algoritmo.

Na segunda etapa, após obter-se a nova representação binária dos dados do arquivo, as informações são transformadas novamente em representação decimal e gravadas em um arquivo no disco com a extensão “.out”.

### 3. Resultados

O algoritmo alcançou o objetivo de comprimir os dados contidos nos arquivos. O tempo de execução da compressão cresce exponencialmente à medida que quantidade de dados aumenta, conforme pode ser verificado na Tabela 2.

Arquivo	Tamanho	Execução 01	Execução 02	Execução 03	Média
49.in	49 bytes	00:00.007	00:00.006	00:00.008	00:00.007
499.in	499 bytes	00:00.049	00:00.046	00:00.058	00:00.051
4999.in	4999 bytes	00:00.940	00:00.837	00:00.826	00:00.868
9999.in	9999 bytes	00:02.911	00:02.967	00:03.102	00:02.993

**Tabela 2: Comparativo dos tempos das execuções**

Em todas as execuções foram utilizados os mesmos recursos computacionais. Uma diferença significativa pode ser notada entre cada execução considerando que os arquivos utilizados nos testes possuem poucos dados. Em todos os arquivos que foram realizados os testes, as taxas de compressão foram muito similares, como pode ser visualizado na tabela 3.

Arquivo	Tamanho	Tamanho após compressão	Taxa de compressão
49.in	49 bytes	34 bytes	30,61%
499.in	499 bytes	342 bytes	31,46%
4999.in	4999 bytes	3.434 bytes	31,31%
9999.in	9999 bytes	6.850 bytes	31,49%

**Tabela 3: Comparativo das taxas de compressão dos arquivos**

Independente da quantidade de dados que estão sendo comprimida, a média da taxa de compressão ficou em 31,22%. Podemos verificar que a taxa de compressão se mantem com poucas alterações a medida que a quantidade de dados aumenta.

### 4. Conclusão

Este trabalho apresentou o desenvolvimento de um algoritmo de compressão de dados baseado na codificação de Huffman. Como citado nos resultados, o objetivo proposto foi alcançado, porém o algoritmo está longe de ser a solução ótima para compressão de dados. Isto fica muito evidente ao verificarmos que o tempo de execução aumenta consideravelmente à medida que aumenta o volume de dados ao mesmo tempo que taxa de compressão se mantém com pouca variação.

Estas deficiências ficam ainda mais evidente se os resultados do algoritmo proposto forem comparados com as soluções existentes no mercado para compressão de dados. Estas deficiências geram algumas propostas para trabalhos futuros. Pode-se citar duas. A primeira é a melhora de performance no algoritmo, diminuindo assim o tempo

de compressão. A segunda sugestão é utilizar a representação binária para mais caracteres dos dados, visto que neste trabalho utilizou-se apenas 4, e também representar de forma binária expressões que se repetem com frequência no texto.

## **References**

ARRUDA, Diego Vidier, GOES, Rodrigo Silva. Programa para compactação de dados utilizando código de Huffman. 2003. Disponível em: <<http://migre.me/bZTxW>>. Acesso em: 20 de novembro de 2012.