INDIAN INSTITUTE OF INFORMATION TECHNOLOGY NAGPUR

Name: Uma

Roll Number: BT20ECE044

Semester: 6

Course: Coding Techniques

Case Study - 2

A Hybrid Compression Algorithm by Using Shannon-Fano Coding and Oring Bits

Reference: Journal of Kerbala University , Vol. 6 No.3 Scientific. 2008

Author:

- Bayadir Al-Himyari
  University of Babylon

Year of Publication: July 2022

Abstract:

Bayadir Al-Himyari proposed a hybrid compression algorithm that combines Shannon-Fano coding with an ORing bit technique. The algorithm first divides the input data into smaller segments and applies Shannon-Fano coding to each segment separately. Then, an ORing bit technique is applied to further compress the encoded data. The ORing bit technique involves XORing the encoded data with a bit sequence generated from the previous encoded segment, which results in a more compact representation of the data and also achieves higher compression ratios and faster encoding and decoding times.

Data compression is a widely used technique for reducing the size of data files without affecting their essential information. The need for data compression arises due to limited storage capacity and the requirement for fast data transfer.

The Shannon-Fano coding technique is a simple yet efficient technique used for data compression. It involves dividing the input data into smaller segments and assigning variable-length codes to each segment based on their probability of occurrence. However, Shannon-Fano coding has its limitations, and the algorithm is known to produce suboptimal compression ratios for certain types of data.

Table(1) illustrates the Shannon-Fano code for a seven-symbol alphabet (the symbols are not shown, only their probabilities).

Table (1) Shannon-Fano Example

	Probab.	Steps	Final code
1.	0.25	1 1	:11
2.	0.20	1 0	:10
3.	0.15	0 1 1	:011
4.	0.15	0 1 0	:010
5.	0.10	0 0 1	:001
6.	0.10	0 0 0 1	:0001
7.	0.05	0 0 0 0	:0000

The lengths of the codewords is equal to $\lceil \log_2(1/p_i) \rceil$, where p_i is the probability.

To overcome the limitations of Shannon-Fano coding, the paper proposes a hybrid algorithm that combines Shannon-Fano coding with the ORing bit technique. The ORing bit technique involves XORing the encoded data with a bit sequence generated from the previous encoded segment. This technique results in a more compact representation of the data, leading to higher compression ratios.

This method starts with a sparse string L1 of size n1 bits. In the first step, L1 is divided into k substrings of equal size. In each substring, all bits are logically ORed, and the results (one bit per substring) become string L2, which will be compressed in step 2. All zero substrings

of L1 are now deleted. In step 2, the same process is applied to L2, and the result is the 4-bit string L3.

Example,

L_1= 0000 | 0000 | 0000 | 0100 | 0000 | 0000 | 0000 |
1000 | 0000 | 0000 | 0000 | 0000 | 0010 | 0000 |
0000 | 0000

L_1 is a 64-bit string divided into 16 substrings of size 4 each. After Oring each 4-bit substring we get the 16-bit string L_2.

L_2= 0001 | 0001 | 0000 | 1000.

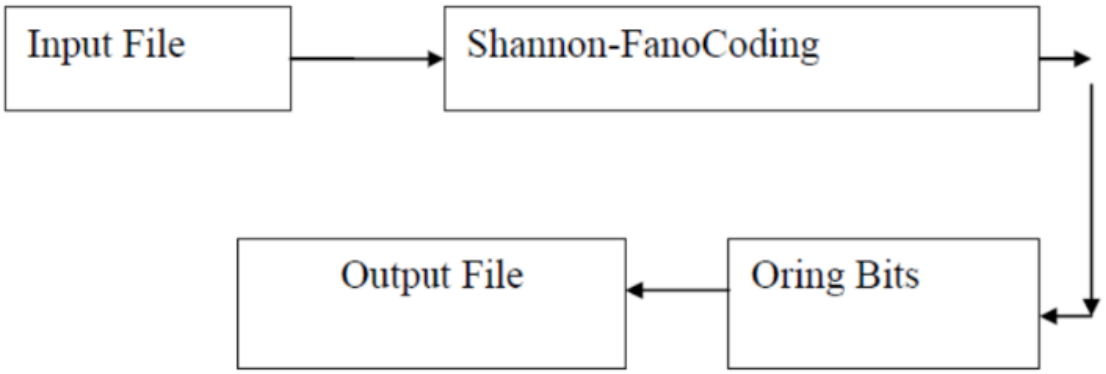
which is short enough so no more compression steps are needed. After deleting all zero substrings

in L_1 and L_2 we end up with the three short strings

L_1 = 0100 | 1000 | 0010, L_2= 0001 | 0001 | 1000

L_3= 1101.

Proposed compression method:-



Results:

File	File size	File size(1)	File size(2)	Compression Ratio
File1	81 byte	38 byte	28 byte	65.432 %
File2	600 byte	295 byte	235 byte	60.883 %
File3	884 byte	200 byte	198 byte	77.602 %
File4	100 byte	48 byte	43 byte	57n%
File5	427 byte	308 byte	308 byte	40.281 %
File6	292 byte	81 byte	81 byte	72.26 %
File7	600 byte	207 byte	207 byte	66.167 %
File8	97 byte	48 byte	48 byte	70.103 %
File9	204 byte	87 byte	87 byte	75 %
File10	250 byte	78 byte	78 byte	76.8 %
File11	296 byte	120 byte	82 byte	72.297 %
File12	436 byte	154 byte	125 byte	71.33 %

Experimental results showed that the proposed algorithm achieved higher compression ratios compared to the standard Shannon-Fano coding technique. The algorithm was also evaluated for its efficiency in terms of encoding and decoding speed, and the results showed that it performed reasonably well in both aspects. Overall, the proposed hybrid compression algorithm shows promise as a viable technique for data compression.

The proposed algorithm's efficiency makes it suitable for use in various applications, including multimedia compression, network transmission, and storage systems.