IJournals: International Journal of Software & Hardware Research in Engineering
ISSN-2347-4890
Volume 4 Issue 11 November, 2016

GOLDBACH CODES ALGORITHM FOR TEXT COMPRESSION

Surya Darma Nasution[1], Mesran*[2]

[1]*Lecturer, STMIK Budidarma, Jl. Sisingamangaraja XII No.338, SitiRejo I, Medan Kota, Kota Medan, Sumatera Utara 20216, Indonesia, darmashadow@gmail.com*

[2]*Lecturer, STMIK Budidarma, Jl. Sisingamangaraja XII No.338, SitiRejo I, Medan Kota, Kota Medan, Sumatera Utara 20216, Indonesia, mesran.skom.mkom@gmail.com*

KEYWORD
Goldbach Codes Algorithm, Algoritm, Compression, Text Compression, Goldbach Algoritm

ABSTRACT
Data compression is converting data in the form of a group of characters into a form of code in order to save storage requirements and data transmission time. There are several factors taken into consideration in choosing the algorithm to be used in the compression of data that required resources (memory, speed, PC), the compression speed, the size of the compression, the amount of redundancy and complexity of the algorithm. Goldbach Codes algorithm is an algorithm which is assumed to use the theory of Goldbach Conjecture is all positive even number greater than 2 is the sum of two primes.

1. INTRODUCTION
Compression is the conversion of data in the form of a group of characters into a form of code in order to save storage requirements and data transmission time. There are several factors taken into consideration in choosing the algorithm to be used in the compression of data that required resources (memory, speed PC), the compression speed, the size of the compression, the amount of redundancy and complexity of the algorithm. If the larger the size of the stored data, the need for large storage media. Therefore, later appeared methods that aim to compress the data in order to save data storage. One of them is the compression of text data to be reduced in size.

Method of compression of text data can be grouped into two major groups, namely Dictionary based, which works by changing the groups of symbols in the data into the codes of a certain length, assuming the codes are generally shorter than symbol set is replaced and Statistical based that compress with a different approach, in which the existing symbols encoded one by one, so that the length of the code output will vary depending on the probability or frequency of occurrence of symbols. Grouping compression method is based on the integrity of the data, which is "Lossy Compression" and "Lossless Compression".

2. THEORY
2.1 DATA COMPRESSION TECHNIQUE
Lossless data compression algorithms are used to reduce the amount of source information to be transmitted in such a way that when compression information is decompressed, there is no loss of information. Lossless compression is possible because most real-world data have statistical redundancy and these algorithms exploit these statistical redundancies to represent data more concisely without losing information.

Lossy data compression is contrasted with lossless data compression. Lossy data compression algorithms do not produce an exact copy of the information after decompression as was present before compression. In these schemes, some loss of information is acceptable. Lossy Compression reduce the file size by eliminating some redundant data that won't be recognized by humans after decoding [2]

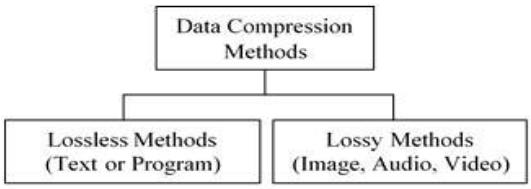

Fig 1. Representation of compression methods

2.2 GOLDBACH CODES ALGORITHM

A prime number is an integer that is not divisible by any other integer (other than trivially by itself and by 1). The integer 2 is prime, but all other primes are odd. Adding two odd numbers results in an even number, so mathematicians have always known that the sum of two primes is even. History had to wait until 1742, when it occurred to Christian Goldbach, an obscure German mathematician, to ask the "opposite" question. If the sum of any two primes is even, is it true that every even integer is the sum of two primes? Goldbach was unable to prove this simple-looking problem, but neither was he able to find a counter-example. He wrote to the famous mathematician Leonhard Euler and received the answer "There is little doubt that this result is true." However, even Euler was unable to furnish a proof, after more than 260 years of research, the Goldbach conjecture has almost, but not quite, been proved. It should be noted that many even numbers can be written as the sum of two primes in several ways. Thus 42 = 23 + 19 = 29 + 13 = 31 + 11 = 37 + 5, 1,000,000 can be partitioned in 5,402 ways, and 100,000,000 has 291,400 Goldbach partitions.

In 2001, Peter Fenwick had the idea of using the Goldbach conjecture (assuming that it is true) to design an entirely new class of codes, based on prime numbers. The prime numbers can serve as the basis of a number system, so if we write an even integer in this system, its representation will have exactly two 1's. Thus, the even number 20 equals 7 + 13 and can therefore be written 10100, where the five bits are assigned the prime weights (from left to right) 13, 11, 7, 5, and 3. Now reverse this bit pattern so that its least-significant bit becomes 1, to yield 00101. Such a number is easy to read and extract from a long bitstring. Simply stop reading at the second 1. Recall that the unary code (a sequence of zeros terminated by a single 1) is read by a similar rule: stop at the first 1. Thus, the Goldbach codes can be considered an extension of the simple unary code.

Goldbach's original conjecture (sometimes called the "ternary" Goldbach conjecture), written in a June 7, 1742 letter to Euler, states "at least it seems that every integer that is greater than 2 is the sum of three primes." This made sense because Goldbach considered 1 a prime, a convention that is no longer followed. Today,his statement would be rephrased to "every integer greater than 5 is the sum of three primes." Euler responded with "There is little doubt that this result is true," and coined the modern form of the conjecture (a form currently referred to as the "strong" or "binary" Goldbach conjecture) which states "all positive even integers greater than 2 can be expressed as the sum of two primes." Being honest, Euler also added in his response that he regarded this as a fully certain theorem ("einganzgewissesTheorema"), even though he was unable to prove it [1].

Table 1. The Goldbach G0 Codes

n	2(n+3)	Primes	Codeword
1	8	3+5	11
2	10	3+7	101
3	12	5+7	011
4	14	3+11	1001
5	16	5+11	0101
6	18	7+11	0011
7	20	7+13	00101
8	22	5+17	010001
9	24	11+13	00011
10	26	7+19	0010001
11	28	11+17	000101
12	30	13+17	000011
13	32	13+19	0000101
14	34	11+23	00010001

The first Goldbach code is designated G0. It encodes the positive integer n by examining the even number 2(n + 3) and writing it as the sum of two primes in reverse (with its most-significant bit at the right end). The G0 codes listed in Table 3.42 are based on the primes (from right to left) 23, 19, 17, 13, 11, 7, 5, and 3. It is obvious that the codes increase in length, but not monotonically, and there is no known expression for the length of G0(n) as a function of n The G0 codes are efficient for small values of n because (1) they are easy to construct based on a table of primes, (2) they are easy to decode, and (3) they are about as long as the Fibonacci codes or the standard binary representation (the β code) of the integers.

3. RESULT & DISCUSSION

Goldbach codes algorithm implementation to compress text data, with the following steps:

a. Read all the characters in the text to calculate the frequency of occurrence of each character, where character who appears most is in the first (n = 1) and so on.
b. Find a codeword for each character to be encoded, by finding prime numbers which can represent the sum of two primes. Primes starting from the number 3, 5, 7 and so on.
c. Codeword only have two "1". Search primes represents the sum stops at "1" the second one, for example:

Even number = 24
1 1 7 5 3
3 1
1 1 0 0 0
The reading is conducted from right to left:
3 + 5 = 24 False
3 + 7 = 24 False
5 + 7 = 24 False
3 + 11 = 24 False
5 + 11 = 24 False
7 + 11 = 24 False
3 + 13 = 24 False
5 + 13 = 24 False
7 + 13 = 24 False
11 + 13 = 24 True

After getting a couple of primes is true then the bit pattern that is obtained is 11000. Then the bit pattern is reversed to the most rear bit to "1", resulting in a pattern of bits 00011.Examples of implementing Goldbach codes algorithm were string: "GOOD MORNING", and this process can be seen at this figure

Before Comression

Char	Freq	Decimal Ascii	Binary Ascii	Bit	Bit x Freq
G	2	71	01000111	8	16
O	3	79	01001111	8	24
D	1	68	01000100	8	8
Space	1	83	01010011	8	8
M	1	77	01001101	8	8
R	1	82	01010010	8	8
N	2	78	01001110	8	16
I	1	73	01001001	8	8
Sum of Bit x Freq					96

After Compression

Char	Freq	n	2 (n+3)	Primes	Codeword	Bit	Bit x Freq
O	3	1	8	3 + 5	11	2	6
G	2	2	10	3 + 7	101	3	6
N	2	3	12	5 + 7	011	3	6
D	1	4	14	3 + 11	1001	4	4
Space	1	5	16	5 + 11	0101	4	4
M	1	6	18	7 + 11	0011	4	4
R	1	7	20	7 + 13	00101	5	5
I	1	8	22	5 + 17	010001	6	6
Sum of Bit x Freq							41

Fig 2: The compression process with Goldbach codes algorithm

a. From the string "GOOD MORNING", then earned a string of bits:"10111111001010100111100101011010001011101"
b. In the decompression process is done reading the bit string obtained in the compression process to Figure 2. Reading a string of bits is done from the smallest index until the last index to continue to add value at the previous index which does not represent the character in Figure 2.

Performance measure is use to find which technique is good according to some criteria. The performance of the compression algorithm can be measured on the basis of different criteria depending upon the nature of the application. The most important thing we should keep in mind while measuring performance is space efficiency. Time efficiency is also an important factor. Since the compression behavior depends on the redudancy of symbols in the source file, it is difficult to measure performance of compression algorithm in general. The performance of data compression depends on the type of data and structure of input source. The compression behavior depends on the category of the compression algorithm :lossy or loosles [7]. Following are some measurements use to calculate the performances of lossless algorithms.

a. Compression Ratio : Compression Ratio is the ratio between the size of the file after compression and the size of the file before compression
   Compression Ratio = Size after compression / Size before compression
b. Compression Factor : Compression Factor is the inverse of compression ratio. It is the ratio between the size of the file before compression and size of the file after compression.
   Compression Factor = Size before compression / size after compression

4. CONCLUSION

In this paper, we have talked about the need for data compression and situations where lossy and lossless data compression useful. Some of the algorithms used for lossless compression briefly described and the conclusions drawn. Goldbach Codes (G0) algorithm is a simple compression algorithm encodes only the positive integer n by turning it into a positive integer even with 2 (n + 3) and then wrote a couple summation primes upside-down. After changing the ASCII code, looking for a codeword, modify binary numbers, which ultimately reduces the file size to a size smaller than before.

5. REFERENCES

[1] D. Salomon, G. Motta, "Handbook Of Data Compression", Springer, Fifth Edition, 2010
[2] R. Singh Brar, B. singh ,"A Survey on Different Compression Techniques and Bit Reduction Algorithm for Compression of Text/Lossless Data", International Journal of Advanced Research in Computer Science and Software Engineering, Vol 3, March 2013
[3] A. Vikram Singh, G. Singh,"A Survey on Different text Data Compression Techniques" ,International Journal of Science and Research (IJSR), Vol 3, July 2014
[4] N. Ziviani, E. Moura, G. Navarro, and R. Baeza-Yates. "Compression: A key for next generation text retrieval systems". IEEE Computer Society, 2000, 33(11), pp. 37- 44. 2000
[5] Z. Peter Buba, G. MakshaWajiga, "Radical Data Compression Algorithm Using Factorization", International Journal of Computer Science and Security (IJCSS), Vol 5, 2011
[6] A. Singh, Y. Bhatnagar, "Enhancement of Data Compression Using Incremental Encoding", International Journal of Scientific & Engineering Research, Vol 3, May 2012
[7] S. Porwal, Y. Chaudhary, J. Joshi, M. Jain, "Data Compression Methodologies for Lossless Data and Comparison between Algorithms", International Journal of Engineering Science and Innovative Technology, Vol 2, March 2013
[8] M. Sharma, "Compression Using Huffman Coding", International Journal of Computer Science and Network Security (IJCSNS), VOL 10, May 2010
[9] R. Kaur, Er. Monica Goyal, "An Algorithm For Lossless Text Data Compression", International Journal of Engineering Research & Technology (IJERT), Vol2 , July 2013
[10] I Made Agus, D. Suarjaya ," A New Algorithm for Data Compression Optimization", International Journal of Advanced Computer Science and Applications (IJACSA), Vol 3, 2012.

