International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 15 (2015) pp 36030-36036

Shannon-Fano Coding for Lossless Data Compression – A Review

E. Thirunavukkarasu

Professor and Head Department of Computer Science and Engineering
Jairupaa College of Engineering Tiruppur, India.
Thirunavukkarasu465@gmail. com

Dr. G. Karuppusami

Dean – Research and Innovations Department of Mechanical Engineering
Sri Eshwar College of Engineering Coimbatore, India
Karuppusami. gks@gmail. com

Dr. P. Ezhilarasu

Associate Professor, Department of Computer Science and Engineering,
Hindusthan College of Engineering and Technology, Coimbatore-641032, India
prof. p. ezhilarasu@gmail. com

Dr. N. Krishnaraj

Associate Professor, Department of Information Technology,
Valliammai Engineering College, Kattankulathur, Chennai 603203, India
drnkrishnaraj@gmail. com

Abstract
In this paper, we discuss Shannon-Fano data compression techniques for two types of input. First, input with similar probability of unique characters is considered. Then, input with different probability of unique characters is considered. Its compression ratio, space savings, and average bits also calculated. Each condition compared with other conditions.

Keywords: Shannon-Fano coding, Compression, Encoding, Decoding.

INTRODUCTION

Data compression defined as the reorganization of data in such a way that, the volume of the resultant data is less than that of the volume of the source data. The decompression technique used to get the original data. After decompression if some data lost, then the compression called as lossy compression. If none of the data lost, then the compression called as lossless compression. The Shannon-Fano coding comes under lossless compression. Each compression technique deals with two important aspects. Those are space complexity and time complexity.

Because of compressed data, transfer of data from source to destination performed with the small amount of time. For instance, if the data size is 100MB and the transfer rate between origin and destination is 20 kbps. The time need for the transfer calculated by the given equation 1.

Time for transfer = Input data / transfer rate (1)
(1 MB = 1024 KB and 1 KB = 8 kb)

Input data
= 100 MB
= 100 * 1024 KB
= 100 * 1024 * 8 kb

Transfer rate
= 20 kbps

So time taken for transfer
= (100 * 1024 * 8) / 20
= 5 * 1024 * 8
= 40960 seconds

If the given data compressed into 20MB, then the time taken for transfer will be 8192 seconds.
If the destination allowed amount of storage is 40GB, then the target machine can store the following number of files by using the equation 2.

Number of files can be stored
= Total amount of storage / Size of the file (2)
(1GB = 1024 MB)
= 40 GB / 100 MB
= 40 * 1024 MB / 100 MB
=409. 6 files

Therefore, the destination system can store 409 files for uncompressed data.
For compressed file, it can store
= 40 * 1024 MB / 20 MB
= 2 * 1024
= 2048 files.

The space and time complexity based on compression ratio. It calculated by using the following equation 3.

Compression ratio = Uncompressed actual data / Compressed Data (3)

= 100 MB / 20 MB

= 5:1

The compressed data takes little storage with faster transfer rate than the actual data.

The files are of many types. It may be a text file, image file, video or audio file. The compression ratio differs for each file types.

The data compression also has some limitations. If the data to be compressed are video data, we need special hardware. For compression and decompression, it takes some amount of time. During compression and decompression, the some data may be lost. The limitations aggregated as

1. Data quality
2. Time for processing
3. Cost for processing

In compression, we have the following types

1. Lossy compression (Compressed data size is less than source data)
2. Lossless compression (Compressed data size is equal to source data).

In this paper, we discuss Shannon-Fano lossless data compression algorithm.

RELATED WORK

Shannon-Fano coding technique was developed independently by Shannon and Fano in 1944. Shannon introduced the concept [1] and later the encoding of the message was implemented by Fano [2]. Mark Nelson and Jean-loup Gailly [1995] described the basics of data compression algorithms. It includes lossless and lossy algorithms [3]. David Salomon [2000] described many different compression algorithms altogether with their uses, limitations, and common usages. He gave an overview on lossless and lossy compression [4]. Khalid Sayood [2000] gave an introduction into the various area of coding algorithms, both lossless and lossy, with theoretical and mathematical background information [5]. Many books [6, 7, 8, 9] published about the data compression techniques.

SHANNON-FANO ENCODING

In the field of data compression, Shannon–Fano coding, named after Claude Shannon and Robert Fano, is a technique for constructing a prefix code based on a set of symbols and their probabilities (estimated or measured) [10].

Algorithm

1. Get the input data.
2. Read the data character by character.
3. Identify unique characters and its occurrences.
4. Find the probability of each unique character.
5. Write the most probable characters to the left of the code table. The least probable characters placed at the right of the code table.
6. Find the result of frequency count difference between, the left part and the right part. If the result very close to zero, Split the list into two parts.
7. The right part assigned the value 1 and the left part assigned the value 0.
8. Apply step six and seven recursively to each of the two halves until we get leaf nodes equal to number of unique characters.

A. ***BASIC EXAMPLE***

If in a message(M), whose length is 100 we have eight unique characters (m1, m2, m3, m4, m5, m6, m7, m8) with occurrences are 30, 30, 10, 10, 5, 5, 5, 5. The probability (P) of each unique character given as (p1, p2, p3, p4, p5, p6, p7, p8) given in the equation 4.

Probability of a character (P) = Occurrence of the character / Total length of the message (4)

The probability of the unique characters (m1, m2, m3, m4, m5, m6, m7, m8) calculated as (p1, p2, p3, p4, p5, p6, p7, p8) using the equation 4 and is given in the coding table as given in the table 1.

Table. 1. Coding table for the message(M) with unique characters(m1, m2, m3, m4, m5, m6, m7, m8).

CHARACTER	m1	m2	m3	m4	m5	m6	m7	m8
OCCURRENCE	30	30	10	10	5	5	5	5
PROBABILITY	0.3	0.3	0.1	0.1	0.05	0.05	0.05	0.05

The probability of each unique character is always between zero and one. Here the highest probability is 0. 3 and the least is 0. 05. Initially the list is having eight unique characters m1, m2, m3, m4, m5, m6, m7, m8. The sum of the probability is one. The grouping done from left to right.

1. m1= 0. 3, the sum of the remaining unique characters (m2-m8) probability = 0. 7. So, the difference = 0. 4(0. 7-0. 3).
2. m1+m2=0. 6 the sum of the remaining unique characters (m3-m8) probability = 0. 4. So, the difference = 0. 2(0. 6-0. 4).

The sum of the probability crosses the half of the total probability. In step one, the difference is 0. 4 and in step two it is 0. 2. The least value and its corresponding group taken. The group (m1-m8) broken into two groups (m1-m2) with assigned value zero and (m3-m8) with the assigned value one. The group (m1-m2) has only two unique characters with equal probability. The left side character m1 assigned the value 0 and the right side character m2 assigned the value 1.

The group (m3-m8) is having six unique characters with the total probability is 0. 4.

1. m3=0. 1, the sum of the remaining unique characters (m4-m8) probability = 0. 3. So the difference = 0. 2(0. 3-0. 1).

2. m3+m4=0. 2, the sum of the remaining unique characters (m5-m8) probability = 0. 2. So the difference = 0. 0(0. 2-0. 2).

The sum of the probability reaches the half of the total probability. In step one, the difference is 0. 2 and in step two it is 0. 0. The least value considered. The group (m3-m8) divided into two groups (m3-m4) with assigned value zero and (m5-m8) with the assigned value one. The group (m3-m4) has only two unique characters with equal probability. The left side character m3 assigned the value 0 and the right side character m4 assigned the value 1.

The group (m5-m8) is having four unique characters with the total probability is 0. 2.

1. m5=0. 05, the sum of the remaining unique characters (m6-m8) probability = 0. 15. So the difference = 0. 1(0. 15-0. 05).
2. m5+m6=0. 1, the sum of the remaining unique characters (m7-m8) probability = 0. 1. So the difference = 0. 0(0. 1-0. 1).

The sum of the probability reaches the half of the total probability. In step one, the difference is 0. 1 and in step two it is 0. 0. The least value considered. The group (m5-m8) broken into two groups (m5-m6) with assigned value zero and (m7-m8) with the assigned value one. The group (m5-m6) and (m7-m8) has only two unique characters with equal probability. The left side character m3, m7 assigned the value 0 and the right side character m6, m8 assigned the value 1. The result shown in Figure 1 and in Table 2.

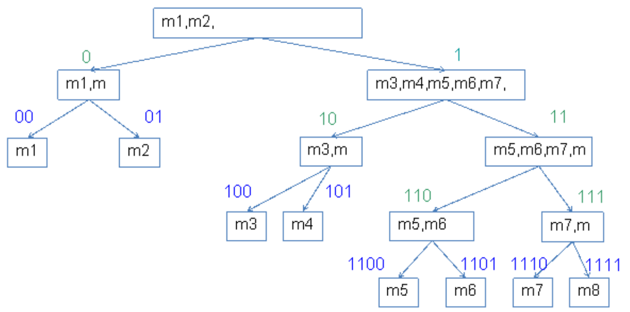

Fig. 1. Shannon-Fano Tree

Table. 2. Shannon-Fano Encoding

Message	**m1**	**m2**	**m3**	**m4**	**m5**	**m6**	**m7**	**m8**
Probability	0. 3	0. 3	0. 1	0. 1	0. 05	0. 05	0. 05	0. 05
Encoding vector	00	01	100	101	1100	1101	1110	1111

The total number of bits needed
= 30 *2 + 30 *2 + 10*3 + 10*3 +5*4 +5*4 + 5*4 +5*4
=60+60+30+30+20+20+20+20
=260 bits

The size of the input as uncompressed
= 100 * 8
= 800 bits

The size of the compressed data using binary coding =100*3=300 bits.

B. ADVANCED EXAMPLE

INPUT

"Dr. Ezhilarasu Umadevi Palani has more than two decades of academic experience in teaching, research, and industry. "
The input placed between " ". The input has 114 characters with 29 unique characters. Each unique character has some occurrences, as shown in table 3.

Table. 3. Shannon-Fano character occurrence for the given input

S. NO	SYMBOL	OCCURRENCE	PROBABILITY
1.	" "(space character)	15	0. 131579
2.	a	13	0. 114035
3.	e	12	0. 105263
4.	i	8	0. 070175
5.	n	7	0. 061404
6.	r	7	0. 061404
7.	d	6	0. 052632
8.	c	6	0. 052632
9.	s	5	0. 04386
10.	h	5	0. 04386
11.	t	4	0. 035088
12.	o	3	0. 026316
13.	m	3	0. 026316
14.	,	2	0. 017544
15.	.	2	0. 017544
16.	l	2	0. 017544
17.	u	2	0. 017544
18.	y	1	0. 008772
19.	g	1	0. 008772
20.	p	1	0. 008772
21.	x	1	0. 008772
22.	f	1	0. 008772
23.	w	1	0. 008772
24.	v	1	0. 008772
25.	z	1	0. 008772
26.	p	1	0. 008772
27.	U	1	0. 008772
28.	E	1	0. 008772
29.	D	1	0. 008772

The initial probability = one. Half of the probability =0. 5.
The sum of first five characters in terms of probability = 0. 482456
The sum of first six characters in terms of probability = 0. 54386
Immediate left value = 0. 482456
Immediate right value=0. 54386
The nearer value to the half of the probability is 0. 482456.
Hence, the first five characters grouped as group 1 with assigned value 0, and last twenty-four characters grouped as group 2 with the assigned value 1.

In the group 1(First five unique characters), the initial probability = 0. 482456. Half of the probability =0. 241228
The first character probability = 0. 131579
The sum of first two characters in terms of probability = 0. 245614
Immediate left value = 0. 131579
Immediate right value=0. 245614
The nearer value to the half of the probability is 0. 245614
Hence, the first two characters grouped as group 1 with assigned value 00, 01 and third, fourth, and fifth characters grouped as group 2 with the assigned value 1.
So space character " " assigned the vector = 000
'a' assigned the vector =001
The group that contains third, fourth and fifth character has the probability 0. 236842. Half of the probability = 0. 118421
The third character probability = 0. 105263
The sum of third and fourth characters in terms of probability = 0. 175438
Immediate left value = 0. 105263
Immediate right value=0. 175438
The nearer value to the half of the probability is 0. 105263
Hence, the third character assigned the value 0 and fourth, and fifth characters are will assign the value 10, 11.
This process continues up to the last unique character. It represented in figure 2, 3 and table 4. The encoding vector for each unique character represented in the table 5.

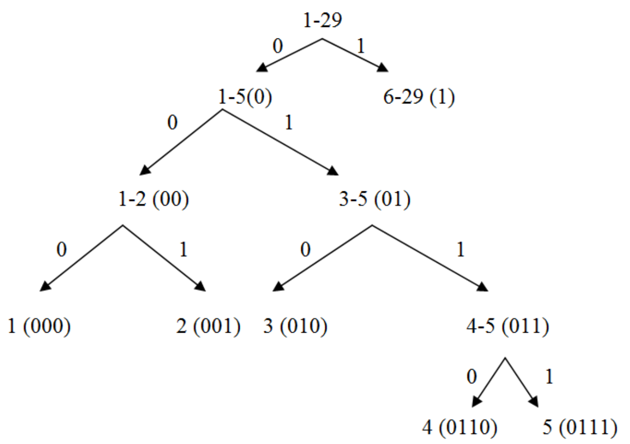

Fig. 2. Formation of groups (1-5) from (1-29)

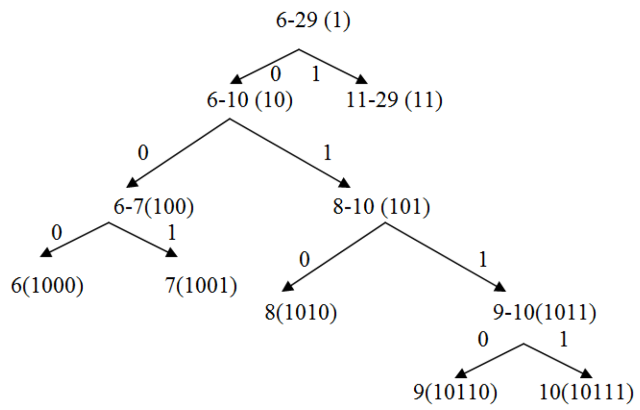

Fig. 3. Formation of groups (1-5) from (1-29)

Table. 4. Formation of groups (11-29) from (1-29)

S. NO	GROUP	ASSIGNED CODE	GROUP I	ASSIGNED CODE	GROUP II	ASSIGNED CODE
1	11-29	11	11-16	110	17-29	111
2	11-16	110	11-12	1100	13-16	1101
3	11-12	1100	11	11000	12	11001
4	13-16	1101	13-14	11010	15-16	11011
5	13-14	11010	13	110100	14	110101
6	15-16	11011	15	110110	16	110111
7	17-29	111	17-22	1110	23-29	1111
8	17-22	1110	17-18	11100	18-22	11101
9	17-18	11100	17	111000	18	111001
10	19-22	11101	19-20	111010	21-22	111011
11	19-20	111010	19	1110100	20	1110101
12	21-22	111011	21	1110110	22	1110111
13	23-29	1111	23-25	11110	26-29	11111
14	23-25	11110	23	111100	24-25	111101
15	24-25	111101	24	1111010	25	1111011
16	26-29	11111	26-27	111110	28-29	111111
17	26-27	111110	26	1111100	27	1111101
18	28-29	111111	28	1111110	29	1111111

The probability and encoding vector of each unique character represented in the table 5.

Table. 5. Encoding vector for each unique characters for the given input

S. NO	UNIQUE CHARACTER	PROBABILITY	ENCODING VECTOR
1.	" "(space character)	0. 131579	000
2.	a	0. 114035	001
3.	e	0. 105263	010
4.	i	0. 070175	0110
5.	n	0. 061404	0111
6.	r	0. 061404	1000
7.	d	0. 052632	1001
8.	c	0. 052632	1010
9.	s	0. 04386	10110
10.	h	0. 04386	10111
11.	t	0. 035088	11000
12.	o	0. 026316	11001
13.	m	0. 026316	110100
14.	,	0. 017544	110101
15.	.	0. 017544	110110
16.	l	0. 017544	110111
17.	u	0. 017544	111000
18.	y	0. 008772	111001
19.	g	0. 008772	1110100
20.	p	0. 008772	1110101
21.	x	0. 008772	1110110
22.	f	0. 008772	1110111
23.	w	0. 008772	111100
24.	v	0. 008772	1111010

25.	z	0. 008772	1111011
26.	p	0. 008772	1111100
27.	U	0. 008772	1111101
28.	E	0. 008772	1111110
29.	D	0. 008772	1111111

The size of the compressed data derived from the table 5. It is given in the table 6.

Table. 6. Size of the compressed data for the given input

S. N O	UNIQUE CHARACTER	OCCURRENCE	ENCODING VECTOR	TOTAL LENGTH
1.	“ ”(space character)	15	000	45
2.	a	13	001	39
3.	e	12	010	36
4.	i	8	0110	32
5.	n	7	0111	28
6.	r	7	1000	28
7.	d	6	1001	24
8.	c	6	1010	24
9.	s	5	10110	25
10.	h	5	10111	25
11.	t	4	11000	20
12.	o	3	11001	15
13.	m	3	110100	18
14.	,	2	110101	12
15.	.	2	110110	12
16.	l	2	110111	12
17.	u	2	111000	12
18.	y	1	111001	6
19.	g	1	1110100	7
20.	p	1	1110101	7
21.	x	1	1110110	7
22.	f	1	1110111	7
23.	w	1	111100	6
24.	v	1	1111010	7
25.	z	1	1111011	7
26.	p	1	1111100	7
27.	U	1	1111101	7
28.	E	1	1111110	7
29.	D	1	1111111	7

The given input after encoding will be
1111111100011011011111110111101110111011011011100110000011011011100000011111011101000011001010111101001100001111100001110111001011101100001011100110110000110100110011000010000110001011100101110001100011110011001000100101010100011001010101100001100111101110000000110100011001010110100011010100000010111011011101010101000011001001111010010000011001110001100001000110101011101100111111010011010100010000101011001000110001010101111101010000010111100100001100111100111100010110110001000111001110110

The total number of bits needed is 489 bits.
The size of the input as uncompressed
= 114 * 8
= 912 bits
The size of the compressed data using binary coding =684 bits.

SHANNON-FANO DECODING

The Shannon-Fano decoding implemented by replacing the encoding vector from the starting of the input. i. e., 1111111 replaced by D and so on. The decoding process stops after step number 114(no of characters).

Input after Encoding:
1111111100011011011111110111101110111011011011100110000011011011100000011111011101000011001010111101001100001111100001110111001011101100001011100110110000110100110011000010000110001011100101110001100011110011001000100101010100011001010101100001100111101110000000110100011001010110100011010100000010111011011101010101000011001001111010010000011001110001100001000110101011101100111111010011010100010000101011001000110001010101111101010000010111100100001100111100111100010110110001000111001110110

Input after Decoding

Step1
D1000110110111111011110111011101101101110011000001101101110000000111110111010000110010101111010011000011111000011101110010111011000010111001101100001101001100110000100001100010111001011100011000111100110010001001010101000110010101011000011001111011100000110100011001010110100011010100000010111011011101010101000011001001111010010000011001110001100001000110101011101100111111010011010100010000101011001000110001010101111101010000010111100100001100111100111100010110110001000111001110110

Step2
Dr1101101111110111101110111011011011100110000011011011100000011111011101000011001010111101001100001111100001110111001011101100001011100110110000110100110011000010000110001011100101110001100011110011001000100101010100011001010101100001100111101110000011010001100101011010001101010000001011101101110101010100001100100111101001000001100111000110000100011010101110110011111101001101010001000010101100100011000101010111110101000001011110010000110011110011110001011011000100011100111011

Step3
Dr.1111101111011101110110110111001100000110110111000000111110111010000110010101111010011000011111000011101110010111011000010111001101100001101001100110000100000110001011100101110001100011110011001000100101010100011001010101100001100111101110000011010001100101011010001101010000001011101101110101010100001100100111101001000001100111000110000100011010101011

1011001111110100110101000100001010110010001100010101011111010100000101111001000011001111001111000101
1011000100011100111 0110

Step4

Dr.E111101110111011011011100110000011011011100000011111011101000011001010111101001100001111100001110
11100101110110000101110011011000011010011001100001
0000110001011100101110001100011110011001000100101010100011001010101100001100111101110000011010001100
10101101000110101000001011101101110101010100001100
1001111010010000011001110001100001000110101011101100111111010011010100010000101011001000110001010101
11110101000001011110010000110011110011110001011011
0001000111001110110

Step 5

Dr.Ez10111011011011100110000011011011100000011111011101000011001010111101001100001111100001110111001
01110110000101110011011000011010011001100001000011
00010111001011100011000111100110010001001010101000
11001010101100001100111101110000011010001100101011
01000110101000001011101101110101010100001100100111
10100100000110011100011000010001101010111011001111
11010011010100010000101011001000110001010101111101
01000001011110010000110011110011110001011011000100
0111001110110

Step 6-114.

Dr.Ezh(10111)i(0110)l(110111)a(001)r(1000)a(001)s(10110)u (111000)
(000)U(1111101)m(110100)a(001)d(1001)e(010)v(1111010)i(0110) (000)P(1111100)a(001)l(110111)a(001)n(0111)i(0110)
(000)h(10111)a(001)s(10110)
(000)m(110100)o(11001)r(1000)e(010)
(000)t(11000)h(10111)a(001)n(0111)
(000)t(11000)w(111100)o(11001)
(000)d(1001)e(010)c(1010)a(001)d(1001)e(010)s(10110)
(000)o(11001)f(1110111)
(000)a(001)c(1010)a(001)d(1001)e(010)m(110100)i(0110)c(1010)
(000)e(010)x(1110110)p(1110101)e(010)r(1000)i(0110)e(010)n(0111)c(1010)e(010) (000)i(0110)n(0111)
(000)t(11000)e(010)a(001)c(1010)h(10111)i(0110)n(0111)g(1110100), (110101)
(000)r(1000)e(010)s(10110)e(010)a(001)r(1000)c(1010)h(10111), (110101) (000)a(001)n(0111)d(1001)
(000)i(0110)n(0111)d(1001)u(111000)s(10110)t(11000)r(1000)y(111001). (110110)

RESULT AND DISCUSSION

The compression ratio, space savings and average bits calculated for the two examples are

Basic Example (8 unique characters)

Compression ratio
= 800/260
= 40:13
=3. 08:1

Space savings
=1-(260/800)
= 1-(13/40)
= 1-0. 325
= 0. 675
= 67. 5%

Average bits
= 260/100
= 2. 6 bits per character

Advanced Example (29 unique characters)

Compression ratio
= 912/489
= 304:163
=1. 87:1

Space savings
=1-(489/912)
= 1-(163/304)
= 1-0. 536
= 0. 464
= 46. 4%

Average bits
= 489/114
=4. 29 bits per character

CONCLUSION

The obtained results show that the input with the similar probability gives better compression ratio, space savings, and average bits than the input with the different probability. The Shannon-Fano code gives better compression ratio, space savings, and average bits as compared with the uncompressed data.

REFERENCE

[1] C. E. Shannon, “A mathematical theory of communication”, Bell System Technical Journal, Volume 27, pp. 379–423, 623–656, 1948.

[2] R. M. Fano, “The transmission of information”, Technical Report 65, Research Laboratory of Electronics, M. I. T., Cambridge, Mass., 1949.

[3] Mark Nelson and Jean-loup Gailly, “The Data Compression Book”, M&T Books, New York, United States of America, 2nd edition, 541 pages, 1995.

[4] David Salomon, “Data Compression: The Complete Reference”, Springer, New York, Berlin, Heidelberg, United States of America, Germany, 2nd edition, 823 pages, 2000.

[5] Khalid Sayood, “Introduction to Data Compression”, Morgan Kaufmann Publishers, Burlington, United States of America, 2nd edition, 600 pages, 2000.

[6] David Salomon and Giovanni Motta, “Handbook of Data Compression”, Springer London, 2000.

[7] J. A. Storer, “Data Compression”, Computer Science Press, Rockville, MD, 1988.

[8] E. Blelloch, "Introduction to Data Compression", Computer Science Department, Carnegie Mellon University, 2002.

[9] Lynch, J. Thomas, "Data Compression: Techniques and Applications", Lifetime Learning Publications, Belmont, CA, 1985

[10] http://en. wikipedia. org/wiki/Shannon-Fano_coding