COMPARATIVE STUDY AND ANALYSIS OF ADAPTIVE REGION BASED HUFFMAN COMPRESSION TECHNIQUES

Utpal Nandi [1] and Jyotsna Kumar Mandal [2]

[1]Dept. of Comp. Sc. & Engg., Academy Of Technology, West Bengal, India
nandi.3utpal@gmail.com

[2]Dept. of Computer Sc. & Engg., University of Kalyani, West Bengal, India
jkm.cse@gmail.com

ABSTRACT

In this paper, the comparative studies of Adaptive Region Based Huffman compression techniques are done. All these techniques use a region formation algorithm that is also discussed. This algorithm is used to form regions whose size is adjusted depending on the ASCII value difference of elements in the file. One of the techniques is Size Adaptive Region Based Huffman Compression (SARBH) where Huffman codes for entire file are obtained after formation of regions to encode elements of the files. Another technique known as Size Adaptive Region Based Huffman Compression with code interchanging (SARBHI) where interchanging of codes are done between the maximum frequency element of a region and maximum frequency element of entire file before elements of that region are compressed. Another variation of the technique is Size Adaptive Region Based Huffman Compression with selective code interchanging (SARBHS) where region wise interchanging of code is done based on an additional condition. Comparisons in terms of compression ratios and compression times are made among these three techniques and also with Region Based Huffman compression technique and classical Huffman technique. The proposed techniques offer better rates of compression for most of the files. Among these techniques, SARBHS is more effective for all most all types of files.

KEYWORDS

Data compression, Huffman tree, Frequency Table (FT), Symbol Code Table (SCT), compression ratio, Region Based Huffman (RBH)

1. INTRODUCTION

Loss-less data compression techniques are used when storing word processing files, database records or spreadsheets where the loss of a single bit of information could be catastrophic. This type of techniques guaranteed to produce an exact duplicate of the input file/ stream after a compress/expand cycle. One of the loss-less data compression technique is Region Based Huffman (RBH) coding [3]. The technique provides minimum length Huffman code to region-wise maximum frequency element by interchanging codes among maximum frequency element of the entire file and that of region. The technique offers better rate of compression than Huffman coding [12]. But, the problem of RBH coding is that if the proper number of region is not chosen, the performance degrades significantly. The modified technique known as Modified Region Based Huffman (MRBH) coding [3] reduces the problem. It selects the value of number of region from a range provided that offers better result. The selection of proper value of number of region is done by a algorithm known as Region Selection Algorithm (RSA). But, the same problem may occur if the proper value of number of region does not lie within the specified range for a file. It is very difficult to find the optimum value of number of region during the encoding the file. To solve the problem, adaptive Region based Huffman

DOI : 10.5121/ijitcs.2012.2402

compression techniques [1, 2] are introduced. The techniques use a region formation algorithm that forms regions of variable size depending on the ASCII values of the symbols. First, the region formation algorithm is discussed in section 2. The three Adaptive Region Based Huffman Compression techniques i.e. SARBH, SARBHI and SARBHS are discussed in section 3. Results of the techniques are given in section 4. Conclusions are done in section 5. References are followed next.

2. The Region Formation Algorithm

It is noticed for most of the files that differences of ASCII values among adjacent group of elements are close to each other. This fact is used in the Adaptive Region Formation (ARF) algorithm where regions are formed using groups of sequence of characters such that the differences among the ASCII values of elements in a region are within a specified value (r). After grouping elements of file into a number of regions, the information of each region are kept by storing the number of elements, smallest ASCII value and the differences among other ASCII values of elements in the region with the smallest ASCII value. After formation of region, each region containing ASCII values are within the specified value except first two values i.e. the number of symbols and the smallest ASCII value in that region. The proposed Adaptive Region Formation (ARF) algorithm is given in Fig 1. For example, let us consider a file / stream – **DFDDFGDGDDDKLLNLKLMNLLMKTQTWUWTVT** (say MSG1). ARF algorithm with specified value(r) =16 is used to form a number of regions of MSG1. MSG1 is grouped into a number of regions in such a way that each region does not have two symbols with ASCII value difference gather than or equal to 16 as shown in Fig. 2. Information of each regions are kept by storing the number of symbol of each region, smallest ASCII value of all the

```
Input: Input file / stream, value of r.
Output: Number of variable length regions.

1.   Initialize, no_of_region=0 and index=1.
2.   Read one element from input file and store it in array X[index]
     and increment index by one.
3.   Calculate lowest and highest ASCII value of elements of already
     read symbols ($s_1,s_2,s_3,\ldots\ldots s_{m+1}$) in array X as high and low
     respectively.
4.   If (high-low)> r , then
4.1  One region is formed with elements m, low, ($s_1$-low), ($s_2$- low),
     ($s_3$-low),…. , ($s_m$- low) where m is the number of elements of the
     region excluding m and low.
4.2  Increment no_of_region by one and save the value of m and low
     in two dimensional array Y as Y[no_of_region] [1]=m and
     Y[no_of_region][2]= low.
4.3  Set Y[no_of_region] [index]=($s_{i-2}$- low) , for i=3 to m+2.
4.4  Release all symbols from the array A except ($s_{m+1}$-m) from
     X[m+3] and shift $s_{m+1}$ to the first position of the array as X[1]=
     $s_{m+1}$ and set index=2.
     End If.
5.   Repeat step 2 to step 4 until end of file / stream.
6.   Stop.
```

Figure 1. Adaptive Region Formation (ARF) algorithm

symbols and ASCII value difference of all the symbols with smallest ASCII value of symbol in the corresponding region as shown in Fig. 3. After formation of regions, compression techniques are applied.

DFDDFGDGDDD	KLLNLKLMNLLMK	TQTWUWTVT
← Region 1 →	← Region 2 →	← Region 3 →

Figure 2. Elements of variable length regions

11,68,0,1,0,0,1,3,0,3,0,0,0	13,75,0,1,1,3,1,0,1,2,3,1,1,2,0	9,84,0,1,0,3,1,3,0,2,0
← Region 1 →	← Region 2 →	← Region 3 →

Figure 3. Representation of variable length regions of MSG1

3. ADAPTIVE REGION BASED HUFFMAN COMPRESSION TECHNIQUES

Three Adaptive Region Based Huffman Compression Techniques i.e. SARBH, SARBHI and SARBHS are discussed in section 3.1, section 3.2 and section 3.3 respectively with examples. All these techniques use ARF algorithm to group file/message stream into variable size regions.

3.1. Size Adaptive Region Based Huffman Compression (S*ARBH)* Technique

Initially, a number of variable length regions (R1 , R2 , R3 . . . Rn) of input file / stream with a specified value (r) are formed using Adaptive Region Formation (ARF) algorithm. The frequencies of all the elements of the input file / stream whose ASCII value lies in the range 0 to r-1 are also calculated. Then, Huffman Codes of all elements whose ASCII value lies in the range 0 to r-1 of input file / stream is calculated. During the encoding process, first two elements (number of element and smallest ASCII value element) of each region are not coded and all other elements (whose ASCII values lie in the range 0 to r-1) are coded by corresponding Huffman codes. For example, let us consider the file/stream MSG1. The ARF algorithm forms regions as shown in Fig. 3. After formation of three regions, frequencies of the numbers are obtained as given in Table 1 and Huffman tree based on the frequency of numbers is shown in Fig. 4. Codes of each numbers are obtained as given in Table 2. During encoding process, for each region first two elements are kept unchanged and all other elements are coded by corresponding Huffman code. The encoded numbers of each three regions are given in Table 3. Therefore, the compressed file / stream will be-11,68,0,10,0,0,10,111,0,111,0,0,0,13, 75,0,10,10,111, 10,0,10,110,111,10,10,110,9, 84,0,10,0,111,10,111,0,110. The calculation of ratio of compression may be done as- Original message size = 33x8 bits = 264 bits, Only Compressed message size (excluding first two numbers) = 61 bits , Frequency Table size = 6x8 bits = 48 bits, Size of first two numbers of three region=3x2x8 bits = 48bits, Total Compressed message size = (61 + 48+ 48) bits = 157 bits , Compression ratio = { (264 – 157) / 264 }X100 % = 40.53 %.

Table 1. FT of numbers

Number	Frequency
0	14
1	10
2	3
3	6

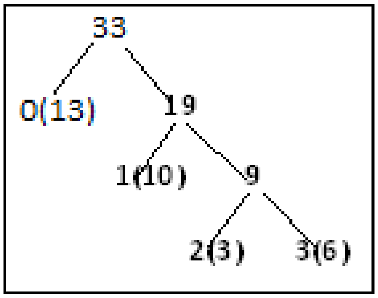

Figure 4. Huffman tree based on Table 1

Table 2. SCTof numbers

Number	Code
0	0
1	10
2	110
3	111

Table 3. Compressed numbers of region 1, 2 and 3 except first two numbers of each region

Region 1		Region 2		Region 3	
Number	**Code**	**Number**	**Code**	**Number**	**Code**
0	0	0	0	0	0
1	10	1	10	1	10
0	0	1	10	0	0
0	0	3	111	3	111
1	10	1	10	1	10
3	111	0	0	3	111
0	0	1	10	0	0
3	111	2	110	2	110
0	0	3	111	0	0
0	0	1	10	-	-
0	0	1	10	-	-
-	-	2	110	-	-
-	-	0	0	-	-

3.2. Size Adaptive Region Based Huffman Compression with code interchanging (S*ARBHI)* Technique

Similar with SARBH, variable length regions (R1 , R2 , R3 . . Rn) of input file / stream are formed using ARF algorithm with a specified value(r). The frequencies of all the numbers of all regions are also obtained whose value lie in the range 0 to r-1 and Huffman Codes of the same are also constructed to obtain the code of each numbers. Instead of that code between maximum frequency element of entire file/stream and the same of each region are interchanged if the code length of that region is larger than maximum frequency element of entire file/stream. During encoding process, for each region first two numbers (number of element and minimum value element) are kept unchanged and all other numbers (whose values lie in the range 0 to r-1) are coded by corresponding Huffman code. For example, let us consider the file/stream MSG1. The ARF algorithm forms regions as shown in Fig. 3. Frequencies of all the numbers in the range 0 to 15 are found as given in Table 1 and Huffman tree based on the frequency of numbers are constructed as shown in Fig. 4 and codes of each number are obtained and shown in Table 2. Maximum frequency number of the entire file /stream (m) is 0. Same of R1 (m1) is 0. As m=m1, no interchange of code is occurred for R1 before compression. Maximum frequency number of R 2 (m2) is 1. As code length of m2 is larger than code length of m**,** interchange of code between m and m2 is occurred and the numbers of R2 are compressed. Maximum frequency number of R 3 (m3) is 0. As m=m3, no interchange of code is occurred for R3 during compression. Compressed numbers of three regions are given below in Table 4. Compressed size of R1, R2 and R3 (excluding first two numbers of each region) are 17, 24, 17 bits respectively. Therefore, the compressed message will be-‘11’,’68’,0,10,0,0,10,111,0, 111,0,0,0‘13’, ’75’,10,0,0,111,0,10,0,110,111,0,0,110,10’9’, ’84’,0,10,0,111,10,111,0,110,0. Size of all regions (excluding first two numbers of each region) is 17+24+17 bits =58 bits. Size of interchange information is 3+2 bits =5 bits. Frequency Table size =6x8 bits =48 bits, size of first two numbers of three region=3x2x8 bits =48 bits. Total Compressed message size=(58+48+48+5) bits=159 bits, Compression ratio={(264–159)/264}X 100 % = 39.77 %. But, there is a limitation of the technique. The code interchanging may increase the compressed message size sometime. To overcome such limitation, another technique is introduced in the following section 3.3.

Table 4. Compressed numbers of region 1, 2 and 3 except first two numbers of each region

Region 1		**Region 2**		**Region 3**	
Number	**Code**	**Number**	**Code**	**Number**	**Code**
0	0	0	10	0	0
1	10	1	0	1	10
0	0	1	0	0	0
0	0	3	111	3	111
1	10	1	0	1	10
3	111	0	10	3	111
0	0	1	0	0	0
3	111	2	110	2	110
0	0	3	111	0	0
0	0	1	0	-	-
0	0	1	0	-	-
-	-	2	110	-	-
-	-	0	10	-	-

3.3. Size Adaptive Region Based Huffman Compression with selective code interchanging (S*ARBHS)* Technique

Similar with S*ARBHI* technique, codes between maximum frequency element of a region and the same of entire file/stream are interchanged. But, one additional condition is checked before interchange of code. Code interchange is not allowed if the overhead (code interchange information size) is more than the benefit (reduction of size for code interchange). For example, let us again consider the file/stream MSG1. The ARF algorithm forms regions as shown in Fig. 3. Frequencies of all the numbers, Huffman tree and codes of each number are obtained like previous technique and shown in Table 1, Fig. 4 and Table 2 respectively. Maximum frequency number of the entire file /stream (m) is 0. Maximum frequency number of R 1 (m1) is 0. As m=m1, no interchange of code is occurred for R1. Maximum frequency number of R 2 (m2) is 1. As code length of m2 is gather than code length of m and overhead (interchange information size = 2bits) is less than benefit (reduction of number of bits=4bits), interchange of code between m and m2 is occurred. Maximum frequency number of R3 (m3) is 0. As m=m3, no interchange of code is occurred for R3. For this example, the encoded numbers of each region are identical with previous SARBHI technique and shown in Table 4. Compressed size of R1, R2 and R3 (excluding first two numbers) are 16, 22 and 16 bits respectively as shown. Therefore, the compressed file/message stream will be-‘11’,’68’,0,10,0,0,10,111,0,111,0,0,0, ‘13’, ’75’ ,10,0,0,111,0,10,0,110,111,0,0, 110,10, ‘9’, ’84’,0,10,0,111,10,111,0,110,0. Similarly, total compressed message size and the compression ratio are calculated as 159 bits and 39.77 % respectively.

4. RESULTS

For experimental purpose, a large number of seven different types of files have been taken. The average compression ratio of each types of file using Huffman technique, RBH coding, MRBH coding and SARBH, SARBHI, SARBHS techniques have been made as shown in Table 5. The compression ratio is calculated by the expression (1 – compressed file/original file) x 100. Here the specified value(r) of ARF algorithm is taken as 128. The graphical representation of the same is shown in Fig. 5. For exe, hybrid and core types of files, all three adaptive Region Based Huffman Compression Techniques offer significantly better rate of compression than Huffman, RBH and MRBH coding techniques. For other types of files, these three compression Techniques offer better rates of compressions than Huffman most of the times and RBH for some values of N. Among the three adaptive Region Based techniques, SARBH offers better compressions for almost all types of files.

Table 5. Comparison of compression ratios in different techniques

File type	Huffman	RBH With N=5	RBH With N=10	RBH With N=20	RBH With N=30	MRBH With Range 10 - 25	SARBH	SARBHI	SARB HS
txt	30.30	30.28	30.43	31.12	31.63	31.40	31.33	31.11	31.20
doc	35.31	35.30	35.33	35.33	35.33	35.34	35.30	35.32	35.32
exe	21.05	21.34	21.63	21.77	21.88	21.82	21.80	21.94	21.88
C++	40.10	40.09	40.12	40.12	40.13	40.14	40.09	40.11	40.11
dll	33.26	33.23	33.21	33.26	33.30	33.40	32.10	33.18	33.26
hybrid	29.87	29.81	29.91	29.95	30.02	29.98	29.97	29.93	30.01
core	10.80	10.78	10.93	10.98	11.13	11.04	11.01	10.99	11.05

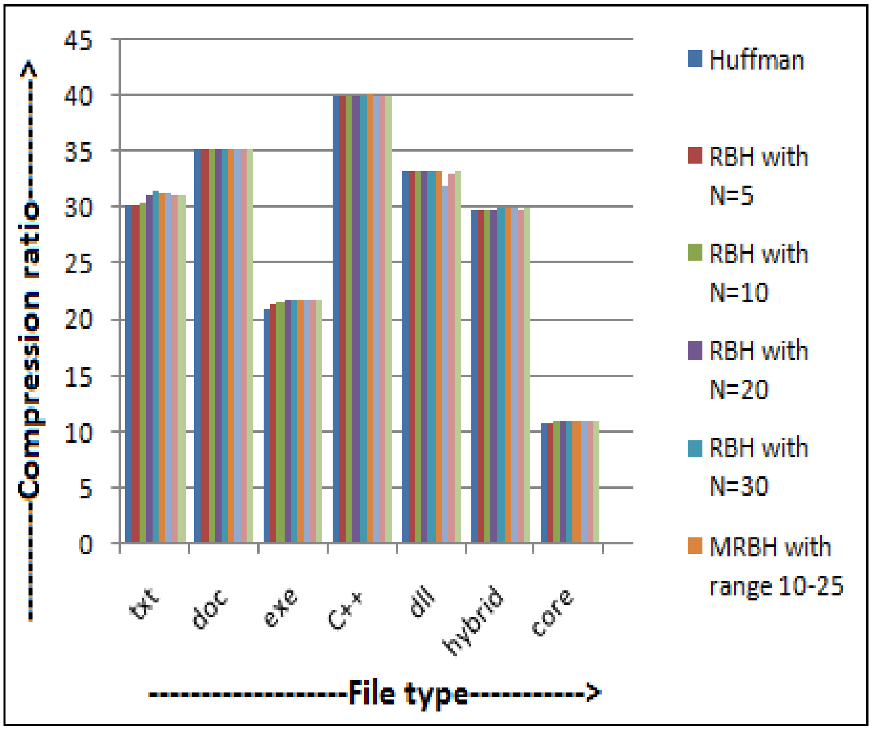

Figure 5. The graphical representation of Comparison of compression ratios in different techniques

5. CONCLUSIONS

The SARBH, SARBHI and SARBHS coding techniques reduce some of the limitations of classical Huffman, RBH and MRBH coding techniques and enhance the performance of SARBH, RBH and MRBH coding by introducing the concept of ARF algorithm which adapts region size based on element's ASCII value difference and region wise interchanging of codes in SARBHI and selective interchanging of codes in SARBHS. The performances in terms of compression ratio of these three techniques are better than Huffman coding for almost all type of files. Even, these techniques offer better results than RBH and MRBH coding for some files. Among the three techniques, SARBHS provides better rate of compression than its counterpart for all most all types of files. The techniques have also a scope of performance enhancement. Instead of region wise single interchanging of code, region wise multiple code interchanging can be done before encoding of elements of each region.

ACKNOWLEDGEMENTS

The authors extend sincere thanks to the department of Computer Science and Engineering and PURSE Scheme of DST, Govt. of India and Academy Of Technology, Hooghly, West Bengal, India for using the infrastructure facilities for developing the technique.

REFERENCES

[1] Nandi, U. & Mandal, J. K., (2012) " Adaptive Region based Huffman Compression Technique With selective code interchanging", Advances in Computing and Information Technology (AISC), Chennai, India, vol. 176, pp. 739-748.

[2] Nandi, U. & Mandal, J. K., (2012) " Size Adaptive Region based Huffman Compression", National Symposium on Emerging Trends in Computer Science (ETCS), Barrackpore, India, pp.

12-14.

[3] Nandi, U. & Mandal, J. K. (2010) " Region based Huffman(RB H) Compression Technique with Code Interchange", Malayasian Journal of Computer Science(MJCS), Malayasia,Vol.23, No. 2, pp. 111-120.

[4] Mandal, J. K. & Kumar, A. (2004) " A Compression Technique Based on Optimality of Huffman Tree (OHT)" , 12th International Conference of IEEE on Advanced Computing and Communications, Ahmedabad, India pp. 589-595.

[5] Ziv, J. & Lempel, A. (1978) "Compression of individual sequences via variable-rate coding", IEEE Transactions on Information Theory. Vol. 24, No.5, pp. 530- 536.

[6] Mandal, J.K. & Gangopadhayay, R. (1995)", Implementation of Two Data Compression Schemes, First International Workshop on Telematics, NERIST, India pp. 154-162.

[7] Reglebati, H. K. (1981) " An Overview of Data Compression Techniques", In IEEE Computer, pp. 71-75.

[8] Ziv, J., & Lempel, A. (1978) "Compression of individual sequences via variable-rate coding," IEEE Transactions on Information Theory, Vol. 24, No.5, pp. 530-536.

[9] Welch & Terry, (1984) " A Technique for High-Performance Data Compression, " IEEE Computer, Vol. 17, No.6, pages 8-19.

[10] Witten, Ian H., Neal, & Radford M., and Cleary, John G., (1987). "Arithmetic Coding for Data Compression," in Communications of the ACM, Vol. 30, No. 6,pp. 520-540.

[11] Ziv, J., and Lempel, A., (1977)."A universal algorithm for sequential data compression," IEEE Transactions on Information Theory, Vol. 23, No. 3, pp.337-343.

[12] Nelson , M., (2008) "The Data Compression Book" ,ed. Second , India, BPB Publications.

Authors

Joytsna Kumar Mandal

M.Tech.(Computer Science, University of Calcutta), Ph.D. (Engg., Jadavpur University), Professor in Computer Science and Engineering, University of Kalyani, Nadia, West Bengal, India. Life Member of Computer Society of India since 1992. 25 years of teaching and research experiences. 8 Scholars awarded Ph.D.; 1 Scholars submitted Ph.D. and 7 scholars are pursuing Ph.D. Total number of publications 228.

Utpal Nandi

M.Sc.(Computer Science, Vidyasagar University),M.Tech.(Computer Science & Enggineering) from University of Kalyani, Nadia, West Bengal, India. Assistant Professor in Computer Science and Engineering, Academy Of Technology, Hooghly, West Bengal, India. Total number of publications 4.