REGION BASED HUFFMAN (RBH) COMPRESSION TECHNIQUE WITH CODE INTERCHANGE

Utpal Nandi[1] , Jyotsna Kumar Mandal[2]
[1]Dept. of Computer Sc. & Engineering, University of Kalyani, Nadia –741235 ,
West Bengal, India, E-mail: nandi_9utpal@yahoo.com
[2] Dept. of Computer Sc. & Engineering, University of Kalyani, Nadia –741235 ,
West Bengal, India, E-mail: jkm.cse@gmail.com

ABSTRACT

There were few research works are continuing to increase the performance of Huffman coding. The proposed paper is based on the new technique region based Huffman to increase the performance of the Huffman coding. The proposed technique divides the input file into a number of regions. Huffman codes are obtained for entire file. For each region, the code between the maximum frequency element of that region and maximum frequency element of entire file are interchanged and the symbols of that region are compressed. This is repeated for each region. Then a small variation of the technique is proposed where instead of interchanging the codes of elements, selection of number of region is done by a proposed algorithm. This modified technique eliminates some limitations of previous proposed technique. Comparisons are made among these two variants with classical Huffman technique.

Keywords: Region Based Huffman (RBH), Compression, Modified Region Based Huffman (MRBH), Region Selection Algorithm (RSA), Huffman Tree, Frequency Table (FT), Symbol Code Table (SCT).

1.0 INTRODUCTION

Compression is the clustering of the data to minimize its representation [1,2,3,4,10]. In order to reduce the data storage requirement and/or the communication cost, there is a need to reduce the redundancy in the data representation, which is, compressing the data. The technique reduces the load and transmission cost on I/O channels in a communication system. A loss-less technique [1,2,3,10] is proposed in this paper based on Huffman tree[2,10], in which, the region wise maximum frequency element gets minimum length code thus overcoming one of the weaknesses of Huffman based techniques. Section 2.0 discusses the weakness of Huffman technique. The proposed RBH coding technique is discussed in section 3.0 in detail. Section 4.0 discusses the weakness of the RBH coding technique. Section 5.0 discusses the region selection process and a region selection algorithm is proposed to overcome the weakness of the proposed RBH coding technique. Section 6.0 includes the proposed MRBH coding technique which uses region selection algorithm. Section 7.0 includes performance analysis. Results are given in section 8.0 and conclusions are drawn in section 9.0.

2.0 LIMITATION OF HUFFMAN CODING

Huffman Coding [2, 10] is based on the frequency of elements of entire file. If an element has maximum frequency, it is assigned with the shortest code. However, we divide a file into a number of regions. It is obvious that in each region the maximum frequency element may not be the maximum frequency element of the entire file and has large code length. If the large codes produced by Huffman coding are used for the elements which have maximum frequency for each region, the size of compressed file increases. For example let us consider a file/stream containing the message as– PQPSQSPSPPQSQPSQSQSQPSSQ RSRSTS (MSG).The frequency of symbols of entire file/stream is given in Table 1.

Table 1 : Frequency of symbols of entire file/stream

Symbol	P	Q	R	S	T
Frequency	7	8	2	12	1

Therefore, the maximum frequency element is S, which gets the minimum code in Huffman coding. Now the same input message stream (MSG) is divided into 3 regions region1 (R1), region2 (R2) and region3 (R3) as shown in Fig.1.

PQPSQSPSPP	QSQPSQSQSQ	PSSQRSRSTS
← region 1(R1) →	← region 2(R2) →	← region 3(R3) →

Fig 1: Region wise symbols

For R1, the frequencies of symbols are as given in Table 2.The maximum frequency symbol is P which is not the maximum frequency symbol of entire file. Therefore, the code length of P is not minimum. For R2, frequencies of symbols are given in Table 3. Similarly the maximum frequency symbol is Q which is not the maximum frequency element of entire file. Therefore, the code length of Q is not minimum.

Table 2 : FT of R1

Symbol	P	Q	R	S	T
Frequency	5	2	0	3	0

Table 3 : FT of R2

Symbol	P	Q	R	S	T
Frequency	1	5	0	4	0

For R3, frequency of symbols are given in Table 4. The maximum frequency symbol is S which is the maximum frequency symbol of entire file.

Table 4: FT of R3

Symbol	P	Q	R	S	T
Frequency	1	1	2	5	1

Therefore, the maximum frequency symbols of R1, R2 and maximum frequency symbol of entire file are not same. The maximum frequency symbols of R1 and R2 can not obtain the minimum length code which may increase the size of compressed file. Therefore, it can be concluded that Huffman coding has some limitations. In this paper a technique is proposed to overcome these limitations in section 3.0.

3.0 THE SCHEME

The proposed coding technique divides the total input file/stream into a number of regions N. The maximum frequency elements for each region are calculated. Huffman codes are obtained based on frequency of elements for entire file/stream. Now for the first region, if the code length of maximum frequency element of that region is larger than the code length of maximum frequency element of entire file/stream, the code between maximum frequency element of that region and maximum frequency element of entire file/stream is interchanged. This interchanged information is attached with the compressed file/stream. The elements of that region are compressed with the changed codes. Otherwise, same symbol code table is used. Similarly, all other regions are compressed repeatedly. The technique is termed here as Region Based Huffman (RBH) Coding. The schematic diagram of RBH coding is shown in Fig 2.

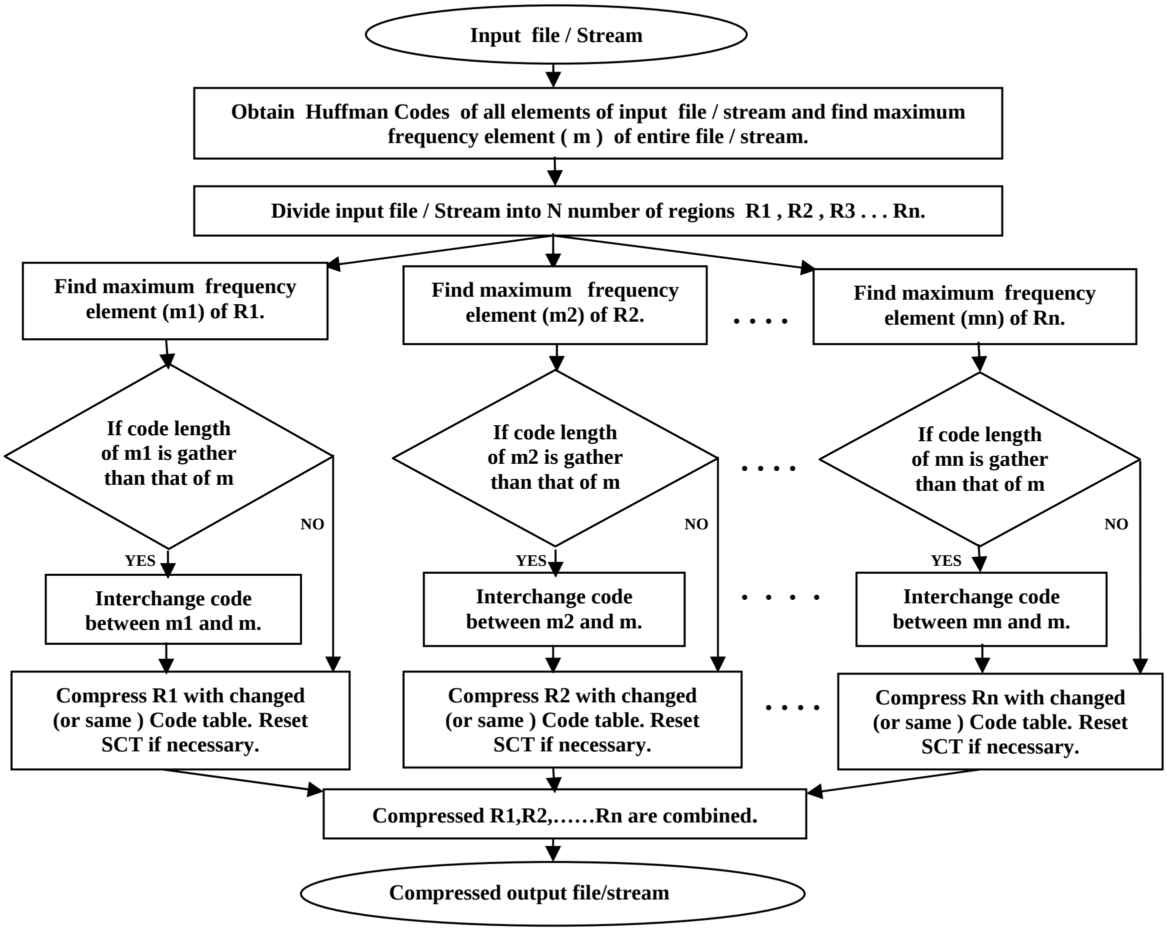

Input file / Stream
Obtain Huffman Codes of all elements of input file / stream and find maximum frequency element (m) of entire file / stream.
Divide input file / Stream into N number of regions R1 , R2 , R3 . . . Rn.
Find maximum frequency element (m1) of R1.
Find maximum frequency element (m2) of R2.
Find maximum frequency element (mn) of Rn.
If code length of m1 is gather than that of m
If code length of m2 is gather than that of m
If code length of mn is gather than that of m
NO
YES
Interchange code between m1 and m.
Interchange code between m2 and m.
Interchange code between mn and m.
Compress R1 with changed (or same) Code table. Reset SCT if necessary.
Compress R2 with changed (or same) Code table. Reset SCT if necessary.
Compress Rn with changed (or same) Code table. Reset SCT if necessary.
Compressed R1,R2,……Rn are combined.
Compressed output file/stream

Fig 2: Schematic diagram of RBH coding

3.1 Example:

Let us consider the same file/stream (MSG) and let N = 3. Frequencies of

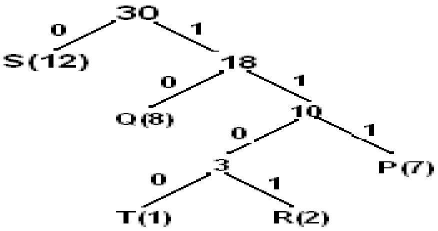


Fig 3 : Huffman tree based on Table 1

symbols of the entire file/stream are counted as given in Table 1. The maximum frequency element is S. A Huffman tree is built based on the frequency of symbols from Table 1 and given in Fig 3. The code of each symbol thus obtained is given in the symbol code table (Table 5).

Table 5 : SCT

Symbol	P	Q	R	S	T
Code	111	10	1101	0	1100

Now the input message stream is divided into 3 regions as given in Fig 1. For R1, the frequencies of symbols are as given in table 2. From table 2, it can be seen that the maximum frequency symbol is P. The maximum frequency symbol under entire message stream is S. Therefore, code of S and code of P are interchanged as the code length of P is greater than the code length of S. The resultant symbol code table is shown in table 6. Then the symbols of R1 are compressed using the changed symbol codes and given in table 7.

Table 6 : Changed SCT for R1

Symbol	P	Q	R	S	T
Code	0	10	1101	111	1100

Table 7 : Compressed symbols of R1

Symbol	P	Q	P	S	Q	S	P	S	P	P
Code	0	10	0	111	10	111	0	111	0	0

Symbol codes are reset as obtained from the Huffman tree. For R2, the frequencies of symbols are as given in table 3. From table 3, it can be seen that the maximum frequency symbol is Q. Code of S and code of Q are interchanged. The resultant symbol code table is shown in table 8. Then the symbols of R2 are compressed using the changed symbol codes and shown in table 9.

Table 8: Changed SCT for R2

Symbol	P	Q	R	S	T
Code	111	0	1101	10	1100

Table 9: Compressed symbols of R2

Symbol	Q	S	Q	P	S	Q	S	Q	S	Q
Code	0	10	0	111	10	0	10	0	10	0

Symbol codes are reset as obtained from the Huffman tree. For R3, the frequencies of symbols are as given in table 4. From table 4, it can be seen that the maximum frequency symbol is S. The maximum frequency symbol of entire message stream (MSG) and R3 are same. Therefore, no interchange of symbol codes occurred. Then the symbols of R3 are compressed using same symbol code table (table 5) as given in Table 10.

Table 10 : Compressed symbols of R3

Symbol	P	S	S	Q	R	S	R	S	T	S
Code	111	0	0	10	1101	0	1101	0	1100	0

Therefore, the compressed message for entire message stream (MSG) will be
0100111101110111000100111100100100111001011010110011000

The effective compression ratio may be obtained by adding the overheads to the compressed size. The calculation may be done as follows: Original message size = 30x8 bits = 180 bits, Frequency Table size = 7x8 bits = 56 bits, Code interchange information size = (3 + 3 + 2) bits = 8 bits , Value of number of region takes 5 bits , Only Compressed message size = 56 bits , Compressed message size including Frequency Table , code interchange information and value of number of region = (56 + 56 + 8 + 5) bits = 125 bits , Compression ratio = { (180 – 125) / 180 }X100 % = 30.55 %.

4.0 WEAKNESS OF RBH CODING

The main disadvantage of RBH coding is the selection of the number of region (N) by the user. If the value of N is not properly chosen, then the compression is not so effective. Proper value of N can be obtained after testing the compression ratio for different values of N and observing for which value of N, the compression ratio is better. But, this is not the task of users. For example let us consider the file containing the same message stream (MSG). It is observed that if the message stream MSG is compressed by taking number of region (N) = 3, then the compression ratio = 30.55 **%.** Now if the message stream is compressed again considering number of region (N) = 5 and compression ratio is reduced to 26.66%. Therefore, the proper value of N must be chosen for better compression of file/stream. Section 5.0 deals with the region selection process to overcome the weakness of the proposed RBH coding technique.

5.0 REGION SELECTION PROCESS

If N is very large, the size of the region decreases. For each region, interchanged information is also attached to the compressed file. For small region size i.e. for large N, the number of interchanged information increases. Therefore, it may increase the size of the compressed file. Again, if N is very small, the potential features of region based technique are not utilized properly and performance will be almost similar with Huffman coding. To solve the problem of selecting the proper value of N, an algorithm termed as Region Selection algorithm has been proposed in section 5.1.

5.1 Region Selection Algorithm (RSA)

Input: Frequency Table, Code Table, input file, L1, L2.
Output: Proper value of N that reduces maximum number of bits.

```
i.    Calculate the total number of bits saved for code interchange as S[N] , for N=L1 to L2.
ii.   Calculate the size of interchange information as Interchange_info[N], for N= L1 to L2.
iii.  Calculate the actual gain of size as Gain[N]=S[N]  -  Interchange_info[N],
      for  N= L1 to L2.
iv.   Return N for N= L1 to L2 such that Gain[N] is maximum.
v.    Stop.
```

The algorithm calculates mathematically the number of bits that can be reduced in RBH coding of a file for a range of values of N (L1-L2) by using code interchange and return the value of N for which it reduces maximum number of bits. L1-L2 can be specified by the user. Therefore, the problem of selecting the proper value of N from the specified range (L1-L2) can be solved using the Region Selection Algorithm. Now the proposed RBH coding is modified using RSA to select the appropriate value of N. The proposed technique is termed here as Modified Region Based Huffman coding and discussed in section 6.0.

6.0 MODIFIED REGION BASED HUFFMAN (MRBH) CODING

The proposed Modified Region Based Huffman (MRBH) coding uses RSA algorithm. The RSA algorithm calculates and returns the proper value of N for a given file/stream. Then RBH coding is used to compress the file/stream using the proper value of N, which may offer better compression. Therefore, the proposed MRBH coding eliminates the problem of RBH coding. It does not require testing the input file for different values of N to find the proper value of N which offers better compression. The task is done by the technique itself.

7.0 PERFORMANCE ANALYSIS

Let the size of Frequency Table in bits be FT, the size of the compressed symbols in bits CS, the size of interchange information in bits XI, the number of bits saved for code interchange XC, the size of the value of Region Number in bits RN and number of region N. The code interchange of each region are occurred or not occurred are indicated by corresponding flag of each region. Consider the size of Flag in bits F. Then in RBH coding, the size of the compressed file = FT + CS + XI + F + RN – XC. But in Huffman coding, the size of the compressed file = FT + CS. Now consider the following two possible cases.

Case 1: No code interchange occurred for any region

As there is no code interchange for each region , the size of interchange information in bits XI =0 , the number of bits stored for code interchange XC =0 and the size of Flag in bits F = N. The size of compressed file = FT + CS + XI + F + RN – XC = FT + CS + N + RN > FT + CS. Therefore, the performance of RBH coding is poorer than Huffman coding. For smaller value of N, N+ RN is very small with respect to FT + CS and the size of compressed file is almost equal to FT + CS. Therefore, in this case the performance of RBH coding is almost similar with Huffman coding for lower value of N.

Case 2: Code interchange occurred

If code interchange occurs in a region, the size of interchange information of that region is equal to the code length of the maximum frequency element of that region. The number of bits saved for code interchange for that region is equal to the difference of code length between maximum frequency element of that region and maximum frequency element of entire file. When XI + F + RN – XC is small (negative), the performance of RBH coding is better. There are three possibilities;

Case 2.1: XI + F + RN = XC

When XI + F + RN = XC i.e. XI + F + RN – XC = 0, the size of compressed file in RBH coding = FT + CS. Therefore, in this case the performance of RBH coding is similar to Huffman coding.

Case 2.2: XI + F + RN < XC

When XI + F + RN < XC i.e. XI + F + RN - XC is negative, the size of compressed file in RBH coding is less than FT + CS. Therefore, in this case the performance of RBH coding is better than Huffman coding.

Case 2.3: XI + F + RN > XC

When XI + F + RN > XC i.e. XI + F + RN - XC is positive, the size of compressed file in RBH coding gather than FT + CS. Therefore, in this case the performance of RBH coding is poorer than Huffman coding.

The performance is dependent on the value of N chosen. The value of N is tuned by using RSA algorithm to obtain the better performance. MRBH coding uses RSA Algorithm to select the proper value of N. Using that value of N, MRBH coding technique compresses the file. The value of N increases the chances of occurrences of the above case 2.2 i.e., XI + F + RN < XC. Therefore, the performance of MRBH coding may be better than Huffman coding for most of the file. But, there may be the possibility that Case 2.1 and Case 2.3 may also occur as there is no control over the context of the file. If there is no code interchange in any region, the size of interchange information in bits XI =0, the number of bits saved for code interchange XC =0 and the size of flag in bits F = N. The size of compressed file = FT + CS + XI + F + RN – XC = FT + CS + N + RN. In this case the Region Selection Algorithm selects lower value of N (i.e. N = L1) from the specified range (L1 – L2). Now N + RN = L1 + RN which is very small with respect to FT + CS. Then, the size of compressed file is almost equal to FT + CS and the performance is almost similar with Huffman coding. Therefore, the performance of MRBH coding may be either better or almost same for most of the file.

8.0 RESULTS

For experimental purpose ten files have been taken as example, and a comparison among Huffman technique, RBH and MRBH coding have been made as shown in table 25. From table 25, it is observed that the ratio of compression of RBH coding is almost the same with Huffman technique for lower values of N (i.e. N=3). For some higher values of N, the ratios of compression of RBH coding are better than Huffman technique.

Table 25: Comparison of compression ratios in different techniques

File Name	%Compression								
	Huffman	RBH With N=3	RBH With N=5	RBH With N=10	RBH With N=15	RBH With N=20	RBH With N=25	RBH With N=30	MRBH With range 10 - 25
Sample.txt	26.84	26.94	27.13	28.12	28.42	28.61	28.27	28.47	28.71
Task.txt	30.31	30.29	30.29	30.44	30.60	31.13	31.41	31.64	31.41
DDA.exe	18.41	18.40	18.57	18.92	18.92	19.10	19.12	19.25	19.37
TRY1.exe	21.09	21.09	21.38	21.67	21.81	21.81	21.75	21.92	21.86
ChipsetCHS.dll	51.25	51.22	51.22	51.20	51.20	51.25	51.32	51.29	51.39
ChipsetARA.dll	24.42	24.49	24.68	24.81	24.81	24.99	24.85	25.30	25.25
Dolly.doc	35.35	35.34	35.34	35.37	35.36	35.36	35.38	35.37	35.38
Resume.doc	35.16	35.16	35.17	35.19	35.19	35.20	35.19	35.20	35.20
Complex.java	35.48	35.48	35.48	35.50	35.51	35.49	35.50	35.52	35.51
Inharit.java	35.24	35.24	35.24	35.24	35.25	35.25	35.27	35.26	35.27

The range of N (10-25) of MRBH coding is chosen after observing the performance of large number of files by using RBH coding. For all ten files, the ratios of compression of MRBH coding are better than Huffman technique.

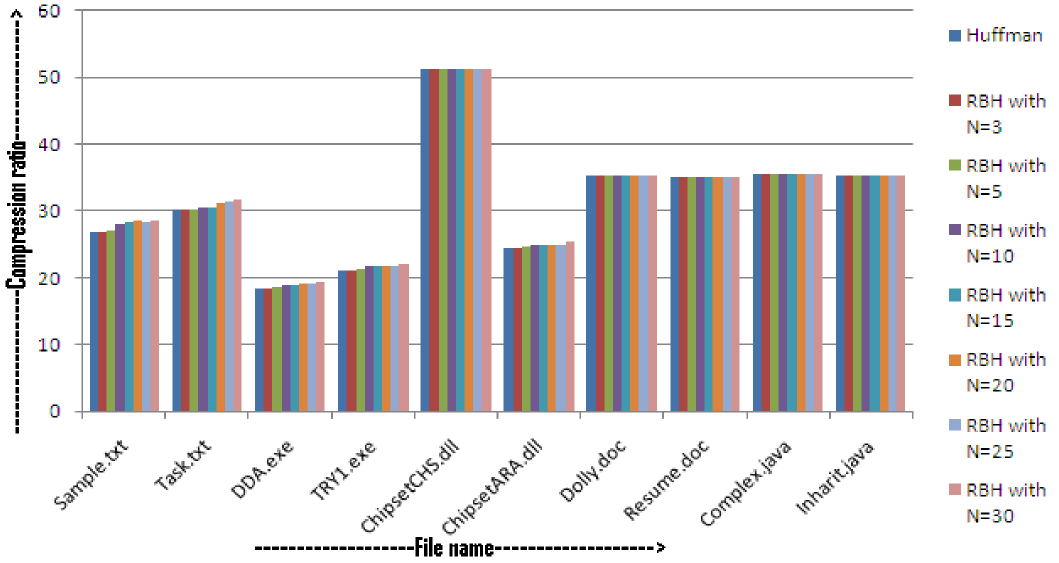


Fig 5: Graphical representation of compression ratios of Huffman, RBH coding.

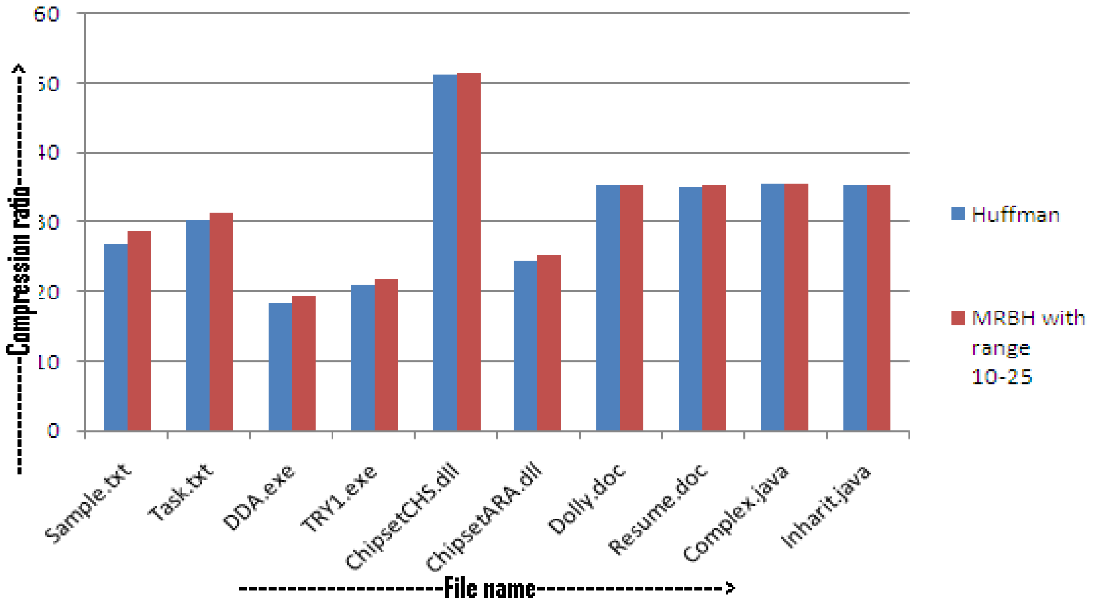
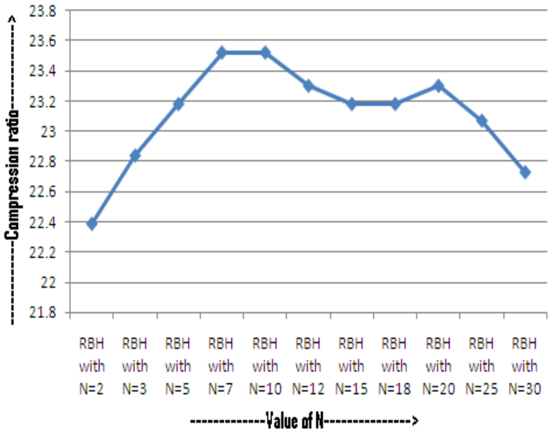


Fig 6: Graphical representation of compression ratios of Huffman, MRBH coding with range of value of N from 10 to 25.

It is also observed how compression ratio changes with different values of N and the size of files as shown in Fig 10 and Fig 11 respectively. From Fig 10, it is observed that the compression ratios of a fixed size file/steam are poor for lower values of N (i.e. N=2, 3). If value of N is increased gradually, the compression ratio goes to optimum value for one or more values of N (i.e. N=7, 10) within a range of values of N (10-25). From Fig 11, the compression ratios of a file/stream generally increases as the size of file/stream increases for fixed value of N (i.e. N=10).



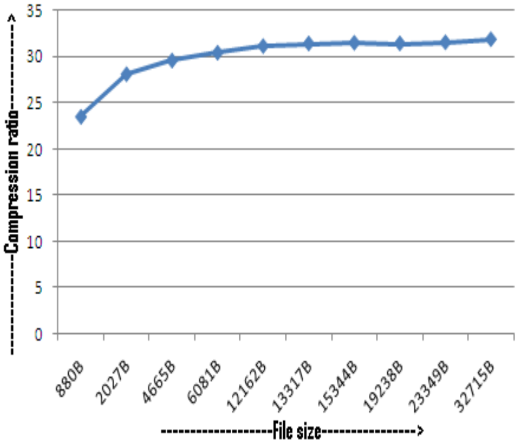


Fig 10: Value of N vs. Compression ratio **Fig 11: File size vs. Compression ratio**

9.0 CONCLUSION

The RBH and MRBH coding are to meet the central idea of enhancing the performance of Huffman coding. The performance of RBH coding depends on the choice of the value of the number of region. The performances of RBH coding are not so effective for lower values of N. For some higher values of N, its performance is better compared to Huffman coding. As MRBH coding uses proper value of number of region (N), the performance of MRBH coding is either better than the Huffman coding or at least almost same with Huffman coding for most of the file. The presented scheme has also a better scope of modification. In MRBH coding, the value of number of region N is chosen from a range of values by using RSA, which may not be optimum. So, there is a scope of modification of the Region Selection Algorithm to obtain the optimal value of number of region. The proposed methods cab also be modified to compress image and the performance can be compared with the existing Region based Image compression using fractals and shape adaptive DCT which is not the scope of this paper.

Acknowledgement: *The authors extend sincere thanks to the department of Computer Science and Engineering, University of Kalyani, Nadia, West Bengal, India for using the infrastructure facilities for developing the technique.*

REFERENCES

[1] Mandal, J. K., Kumar, A., “A Compression Technique Based on Optimality of Huffman Tree (OHT)”, in Proceedings of 12th International Conference of IEEE on Advanced Computing and Communications - ADCOM-2004, December 15-18, 2004, Ahmedabad, India, pp. 589-595.

[2] Huffman, D.A., “A method for the construction of minimum-redundancy codes,” in *Proceedings of the IRE*, Vol. 40, No. 9, September 1952, pp. 1098-1101.

[3] Reglebati, H. K.,” An Overview of Data Compression Techniques,” in *IEEE Computer*, April 1981,pp. 71-75.

[4] Mandal, J.K. and Gangopadhayay, R., “Implementation of Two Data Compression Schemes”, in *Proc. First International Workshop on Telematics*, NERIST, India, 1995, pp. 154-162.

[5] Nelson, M., “LZW Data Compression,” in *Dr. Dobb’s Journal*, Vol. 14, No. 10, October 1989, pp. 29-37

[6] Ziv, J., and Lempel, A., “Compression of individual sequences via variable-rate coding,” in *IEEE Transactions on Information Theory*, Vol. 24, No.5, September 1978, pp. 530- 536.

[7] Welch, Terry, “A Technique for High-Performance Data Compression,” in *IEEE Computer*, Vol. 17, No.6, June 1984, pages 8-19.

[8] Witten, Ian H., Neal, Radford M., and Cleary, John G., “Arithmetic Coding for Data Compression,” in *Communications of the ACM*, Vol. 30, No. 6, June 1987, pp. 520-540.

[9] Ziv, J., and Lempel, A., “A universal algorithm for sequential data compression,” in *IEEE Transactions on Information Theory*, Vol. 23, No. 3, May 1977, pp.337-343

[10] Nelson , M., “*The Data Compression Book*” ,Second edition, India, BPB Publications, 2008.

[11] Kanitkar , Y., “*C project* ”, Second edition, India, BPB Publications, 2002.

[12] Belloulata, K.; Stasinski, R.; Konrad, J,; “Region based Image compression using fractals and shape adaptive DCT,” in proceeding of IEEE international conference on image processing, 1999,icip99. October 24-28, 1999, vol.2, ISBN: 0-7803-5467-2, pp.815-819.

BIOGRAPHY

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. 20 years of teaching and research experiences. 3 Scholars awarded Ph.D.; 3 Scholars submitted Ph.D. and 6 scholars are pursuing Ph.D. Total number. of publications 94.

Utpal Nandi, M.Sc.(Computer Science, Vidyasagar University), M.Tech.(Computer Science & Engg.) from University of Kalyani, Nadia, West Bengal, India.