Canadian Journal of Pure and Applied Sciences
Vol. 10, No. 1, pp. 3795-3801, February 2016
Online ISSN: 1920-3853; Print ISSN: 1715-9997
Available online at www.cjpas.net

RECURSIVE APPROACH FOR MINIMUM REDUNDANT CODE

Om Parkash and *Priyanka Kakkar
Department of Mathematics, Guru Nanak Dev University, Amritsar- 143005, India

ABSTRACT

The present study deals with the investigations of the recursive versions of well known code word lengths due to Campbell (1965) and Kapur (1988). It has been proved that the recursive codes for both the lengths are better than their original code from redundancy point of view and the comparison criterion is the average redundancy taken over the set of all n-tuple distributions. Furthermore, it is shown that when symbol probabilities are arranged in descending order, the recursive versions of both the codes provide much better results. The methodology adopted for this recursive approach involves the use of programming through Matlab and Simulation techniques.

Keywords: Kraft's inequality, entropy, source coding, uniquely decipherable code, redundancy.

INTRODUCTION

It is well known phenomenon that while dealing with source coding, the code designer is usually supposed to be concerned with the reduction in the code rate between the source and the destination whereas an important problem in communication theory is to find the lengths of the prefix-free code words which minimize the redundancy for a given source. The absolute redundancy is defined as the number of bits used to transmit a message lesser the number of bits of the actual information in the message. This idea comes from the fact that in the entire literature of source coding theorems, the mean codeword length is lower bounded by the entropy of the source and it can never be less than the entropy of the source but can be made closer to it.

The main objective in the theory of source coding is to encode the source that produces symbols $X=\{x_i, i=1,2,...,n\}$ with probabilities $P=\{p_i, i=1,2,...,n\}$. The symbols $x_i, i=1,2,...,n$ are encoded to a codeword $c_i, i=1,2,...,n$ of length $l_i, i=1,2,...,n$ using the D letters of the alphabet. By making use of Kraft's (1949) inequality, which is necessary and sufficient condition for the code to be uniquely decipherable given by

$$\sum_{i=1}^{n} D^{-l_i} \le 1, \tag{1.1}$$

Shannon (1948) investigated the first source coding theorem and proved that the entropy of the source provides a lower bound to the average number of code symbols needed to encode each source symbol and proved the following result:

$$H(P) \le L < H(P)+1 \tag{1.2}$$

where $H(P) = -\sum_{i=1}^{n} p_i \log_D p_i$ is a Shannon's entropy and $L=\sum_{i=1}^{n} p_i l_i$ is the mean codeword length.

In case of Shannon's code, $l_i = \lceil -\log_D p_i \rceil, i=1,2,...,n$.

Later, Campbell (1965) and Kapur (1988) proved the source coding theorems for their own exponentiated mean codeword lengths in the form of following inequalities

$$H_\alpha(P) \le L_\alpha < H_\alpha(P)+1 \tag{1.3}$$

and

$$H_\alpha(P) \le L^\alpha < H_\alpha(P)+1, \tag{1.4}$$

respectively where

$L_\alpha = \frac{\alpha}{1-\alpha} \log_D \left(\sum_{i=1}^{n} p_i D^{\frac{l_i(1-\alpha)}{\alpha}} \right), \alpha \ne 1, \alpha > 0$ is Campbell's (1965) mean codeword length,

$L^\alpha = \frac{1}{\alpha-1} \log_D \left(\frac{\sum_{i=1}^{n} p_i^\alpha D^{l_i(\alpha-1)}}{\sum_{i=1}^{n} p_i^\alpha} \right), \alpha \ne 1, \alpha > 0$ is Kapur's (1998) mean codeword length and $H_\alpha(P) = \frac{1}{1-\alpha} \log_D \sum_{i=1}^{n} p_i^\alpha, \alpha \ne 1, \alpha > 0$ is Renyi's (1961) measure of entropy. The lengths used in Campbell (1965) code and Kapur (1998) code are given by

*Corresponding author e-mail: priyanka_kakkar85@yahoo.com

$$l_i^C = \left\lceil -\log_D \frac{p_i^\alpha}{\sum_{i=1}^{n} p_i^\alpha} \right\rceil, i = 1, 2, ..., n$$

and $l_i^K = \lceil -\log_D p_i \rceil, i = 1, 2, ..., n$ respectively.

Chapeau-Blondeau *et al*. (2011) have described a practical problem of source coding by assigning a specific role to Renyi (1961) entropy and investigated an important relation stressing that Renyi's entropy emerges at an order α differing from the traditional Shannon's entropy. Another interesting extension to the source coding theorem based on Shannon entropy was presented by Chapeau-Blondeau *et al*. (2010) and motivated by the development, generalized the same extension for Renyi's entropy. Moreover, the authors demonstrated another lower bound realized by the Tsallis (1988) entropy to the generalized codeword length and also expressed the optimal codeword lengths from the escort probability distributions. Parkash and Kakkar (2012) investigated and introduced two new mean codeword lengths $L(\alpha, \beta)$ and $L(\beta)$ by studying the desirable properties of a measure of typical code lengths and consequently, proved two new noiseless coding theorems subject to Kraft's inequality. Some other related work concerned with the coding theory have been provided by Sharma and Raina (1980), Ramamoorthy (2011), Tu *et al*. (2011) and Koski and Persson (1992) etc.

Recently, Baer (2011) provided some new lower and upper bounds for compression rate of binary prefix codes optimized over memory less sources with the objective to explore exponential average length, maximum point wise redundancy and exponential average point wise redundancy. With the fulfilment of their objectives, the author provided necessary and sufficient conditions for the shortest codeword to be a specific length. Ye and Yeung (2002) took the modified version of the Shannon code whereas Drmota and Szpankowski (2004) proposed a generalized Shannon code and minimized the maximum redundancy to prove its optimality.

Mohajer *et al*. (2012) studied the redundancy of Huffman (1952) code particularly for the sources for which the probability of one of the source symbols is known. For memory less sources with a large alphabet size, Narimani *et al*. (2013) studied the performance of optimal prefix-free encoding, the findings of which made the study of redundancy of the Huffman code for almost all sources with a large alphabet size whereas Narimani *et al*. (2014) compared the performance of Shannon code with that of corresponding well known Huffman code. Related with the above study of redundancy, Parkash and Kakkar (2014) obtained the optimum probability distribution with which the messages should be delivered so that the average redundancy of the source is minimized by considering the case of various generalized mean codeword lengths.

It is generally felt that for efficient source coding, one has to minimize the redundancy prevailing in the system. The present communication is a step in the direction providing recursive versions of various existing codes. To achieve this goal, we make use of the necessary and sufficient condition for unique decipherability of a code due to Kraft's (1949) in which Kraft's sum is generally less than one and it can be made closer to 1 if we reduce the length of some code words in the mean codeword length. In the sequel, we have considered the well known codes due to Campbell (1965) and Kapur (1998) to justify our claim.

2. RECURSIVE VERSIONS OF CAMPBELL AND KAPUR CODE

In the process to find the recursive version of Campbell's (1965) code, we consider the set of constraints

$$D^{-l_i} \le \frac{p_i^\alpha}{\sum_{i=1}^{n} p_i^\alpha}, \ 1 \le i \le n \tag{2.1}$$

which leads to Kraft's inequality when summed over i, $1 \le i \le n$.

In order to introduce recursivity into Campbell's code, we consider the following optimization problem:

Minimize the codeword length

$$L_\alpha = \frac{\alpha}{1-\alpha} \log_D \left(\sum_{i=1}^{n} p_i D^{\frac{l_i(1-\alpha)}{\alpha}} \right), \ \alpha \ne 1, \ \alpha > 0 \tag{2.2}$$

subject to constraint

$$\sum_{j=1}^{i} D^{-l_j} \le \sum_{j=1}^{i} P_j', \ 1 \le i \le n \tag{2.3}$$

where $$P_j' = \frac{p_j^\alpha}{\sum_{i=1}^{n} p_i^\alpha} \tag{2.4}$$

From (2.3), we have

$$D^{-l_i} \le P_i' + \sum_{j=1}^{i-1} \left(P_j' - D^{-l_j} \right), \ 1 \le i \le n$$

which further gives

$$l_i \ge -\log_D \left(P_i' + \sum_{j=1}^{i-1} \left(P_j' - D^{-l_j} \right) \right), \ 1 \le i \le n \tag{2.5}$$

Keeping in view (2.5), we define the Recursive Campbell code as follows:

$$l_i^{RC} = \left\lceil -\log_D \left(P_i' + \delta_i \right) \right\rceil, \ 1 \le i \le n \tag{2.6}$$

where $\delta_1 = 0$ and $\delta_i = \sum_{j=1}^{i-1} \left(P_j' - D^{-l_j} \right), \ i > 1$.

It is to be noted that $\left(l_1^{RC}, l_2^{RC}, ..., l_n^{RC} \right)$ is the suboptimal solution of (2.2) and hence of original Campbell (1965) problem.

Keeping in view the constraint (2.3), we have $\delta_i \ge 0$ for $1 \le i \le n$ and this gives

$$l_i^{RC} \le l_i^C . \tag{2.7}$$

From (2.7), we further have

$$R\left(L_\alpha^{RC},P\right) \le R\left(L_\alpha^{C},P\right) \tag{2.8}$$

where $R\left(L_\alpha^{RC},P\right)$ is the redundancy of Recursive Campbell code and $R\left(L_\alpha^{C},P\right)$ is the redundancy of Campbell code for any distribution P. Thus

$$0 \le R\left(L_\alpha^{RC},P\right) < 1 . \tag{2.9}$$

In case of recursive version of Campbell code, Kraft's inequality is satisfied only when l_i, $1 \le i \le n$ satisfies (2.5). The motivation behind the approach used is to make the Kraft's sum closer to $1=\sum_{j=1}^{n} P_j{}'$. Thus, from (2.8) and (2.9), we have $0 \le R\left(L_\alpha^{C},P\right) - R\left(L_\alpha^{RC},P\right) < 1$

But, this inequality does not give us an idea about the actual difference between redundancy of Campbell code and Recursive Campbell code. To compare the performance of these two codes, the criterion is redundancy and we assume it as a random variable following uniform distribution on the set of all sources having n symbols and use its expected value for evaluating the performance of codes. The performance of a particular code is considered to be better if its average redundancy is lower as compared to the average redundancy of other codes and we consider a uniform distribution because of the reason that all sources are assumed to be equally important.

An important factor to be considered while evaluating the performance of code is the order of probabilities, that is, in what order should the probabilities be arranged while computing Recursive Campbell (RC) code. So, we consider another two codes, the Recursive Campbell Ascending (RCA) code and the Recursive Campbell Descending (RCD) code. When the RC code is applied on the probabilities arranged in ascending order, we call it RCA code and when applied on the probabilities arranged in descending order, we call it RCD code. We will now compare the average redundancy of RCA, RCD and RC code with the Campbell code for the purpose of finding better results.

For comparison purposes, we will make use of simulation technique through Matlab. First of all, we will generate the n tuple distributions and for generating a single n tuple distribution, we will use the following steps in the Matlab (Devroye, 1986.).

1. Generate $n-1$ values $y_1, y_2, ..., y_{n-1}$ from the uniform distribution using the command rand $[1,n]$.
2. Let $temp = 0$ and $i = 1$.
3. For $i = 1$ to $n-1$

   $$p_i = (1-temp)\left(1-\sqrt[n-i]{1-y_i}\right)$$

   $$temp = temp + p_i .$$
4. Then $p_n = 1 - \sum_{j=1}^{n-1} p_j$.
5. $(p_1, p_2, ..., p_n)$ is the required output.

In this way, different $n-$tuple distributions can be generated, and by the use of these distributions, we calculate measure of entropy and mean codeword lengths for each code (Campbell, RC, RCA, RCD code) providing the computations of redundancy and consequently, average redundancy for each code.

Figure 1 shows the average redundancy as a function of n for each of the four codes for $n \le 200$ and $\alpha = 2$. For a single value of n, average redundancy is calculated for 10^4 random $n-$tuple distributions which were generated through simulation as mentioned in the above steps. From figure 1, it is clear that all the recursive versions of Campbell code are better than Campbell code. Further, results are even better if the probabilities are arranged in descending order, that is, RCD code, thus improving the performance of Recursive Campbell code.

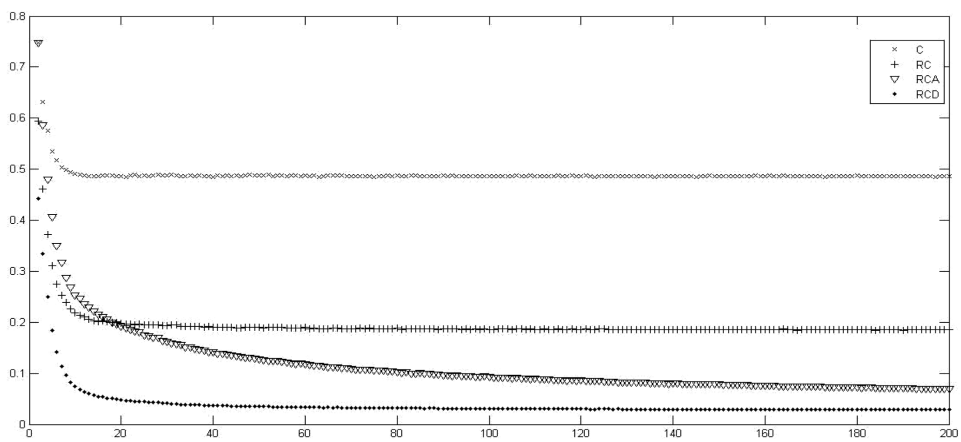

Fig. 1. Average redundancy as a function of n for each of the four versions of Campbell's code for 10^4 random $n-$tuple distributions and $n \le 200$, $\alpha = 2$

For each of the four codes, average redundancy for $n = 200$ for 10^4 random $n-$tuple distributions is shown in Table 1.

Table 1. Average redundancy in case of Campbell code and its recursive versions for $n = 200$, $\alpha = 3$ and 10^4 random $n-$tuple distributions

Campbell Code	Recursive Campbell Code (RC)	Recursive Campbell Ascending Code (RCA)	Recursive Campbell Descending Code (RCD)
0.4820	0.2337	0.1054	0.0229

Further, from Figure 2 and 3, it is observed that curves get distorted if we take lesser number of $n-$tuple probability distributions but the basic result remains the same, that is, all the recursive versions of Campbell code are better than Campbell code and results are even better if the probabilities are arranged in descending order improving the performance of recursive Campbell code. In Figure 2, 10^3 random $n-$tuple distributions are taken and in Figure 3 10^2 random $n-$tuple distributions are considered.

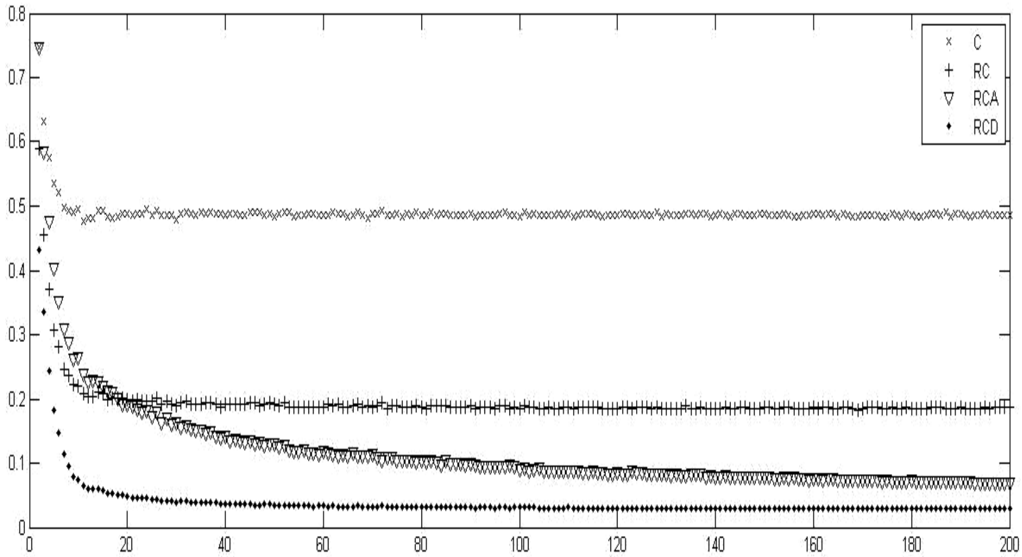

Fig. 2. Average redundancy as a function of n for each of the four versions of Campbell's code for 10^3 random $n-$tuple distributions and $n \leq 200$, $\alpha = 2$

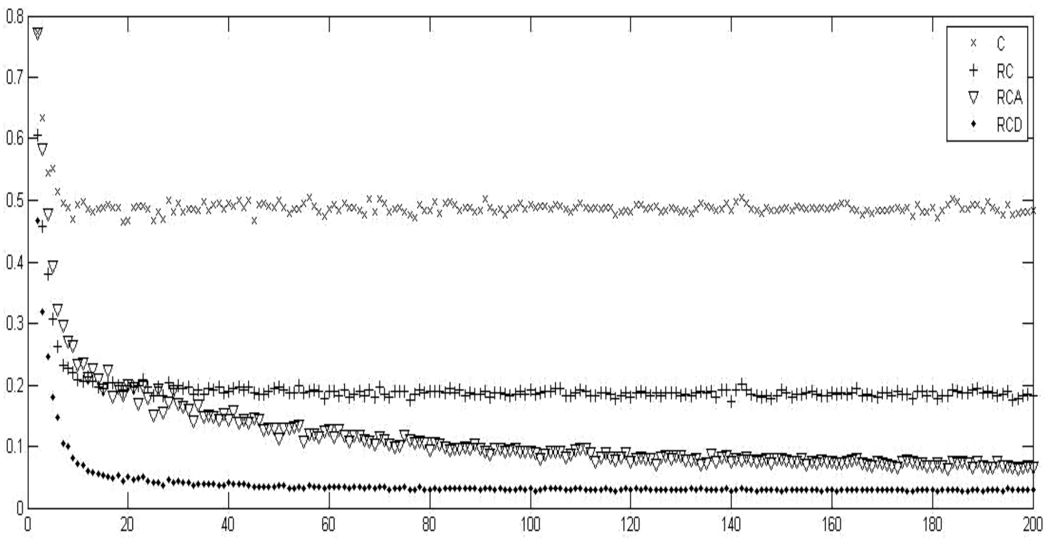

Fig. 3. Average redundancy as a function of n for each of the four versions of Campbell's code for 10^2 random $n-$tuple distributions and $n \leq 200$, $\alpha = 2$

Figures 1, 2 and 3 reveal that the greater the number of $n-$tuple probability distributions, better is the vision about the recursivity.

On similar lines as in the case of Campbell's (1965) code, if we introduce recursivity into Kapur's (1998) code, we get the following results

$$l_i^{RK} = \lceil -\log_D (p_i + \delta_i) \rceil, \ 1 \le i \le n \quad (2.10)$$

where

$\delta_1 = 0$ and $\delta_i = \sum_{j=1}^{i-1} \left(p_j - D^{-l_j} \right), \ i > 1$.

Here also, we compare the performance of four codes Kapur (1998) (K) code, Recursive Kapur (RK) Code, Recursive Kapur Ascending (RKA) Code and Recursive Kapur Descending (RKD) Code by taking into consideration average redundancy of each code. Figure 4 shows the average redundancy as a function of n for each of the four different versions of Kapur's codes for $n \le 200$ where average redundancy is calculated for 10^4 random $n-$tuple distributions for a single value of n. From Figure 4, it is clear that all recursive versions of Kapur code are much better than their original code and further RKD code gives best results. For each of the four versions of Kapur's code, average redundancy for $n = 200, \alpha = 2$ for 10^4 random $n-$tuple distributions is shown in Table 2.

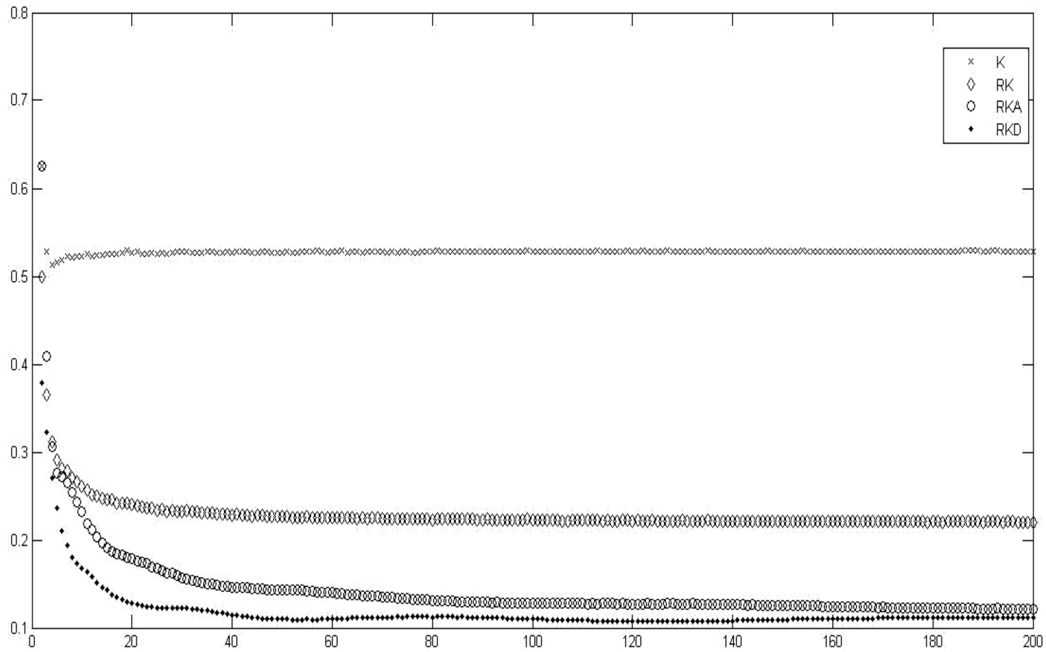

Fig. 4. Average redundancy as a function of n for each of the four versions of Kapur's code for 10^4 random $n-$tuple distributions and $n \le 200, \alpha = 2$

Table 2. Average redundancy in case of Kapur code and its recursive versions for $n = 200$, $\alpha = 2$ and 10^4 random $n-$tuple distributions

Kapur Code	Recursive Kapur Code (RK)	Recursive Kapur Ascending Code (RKA)	Recursive Kapur Descending Code (RKD)
0.5285	0.2203	0.1211	0.1118

Similarly, as in case of Campbell's code, it may be observed that the main difference between the recursive versions of Kapur's (1988) code and the original Kapur code will remain the same for any number of $n-$tuple probability distributions but one gets clear picture about recursivity if the number of $n-$tuple probability distributions are greater as can be seen from Figures 4, 5 and 6.

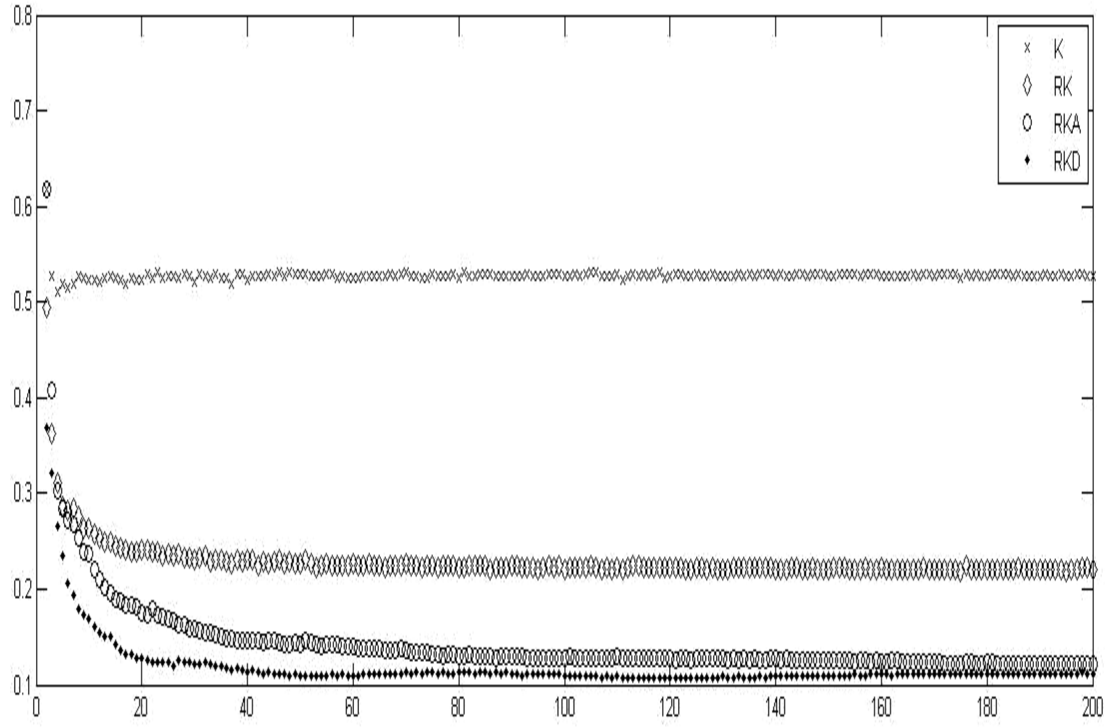

Fig. 5. Average redundancy as a function of n for each of the four versions of Kapur's code for 10^3 random $n-$tuple distributions and $n \leq 200$, $\alpha = 2$

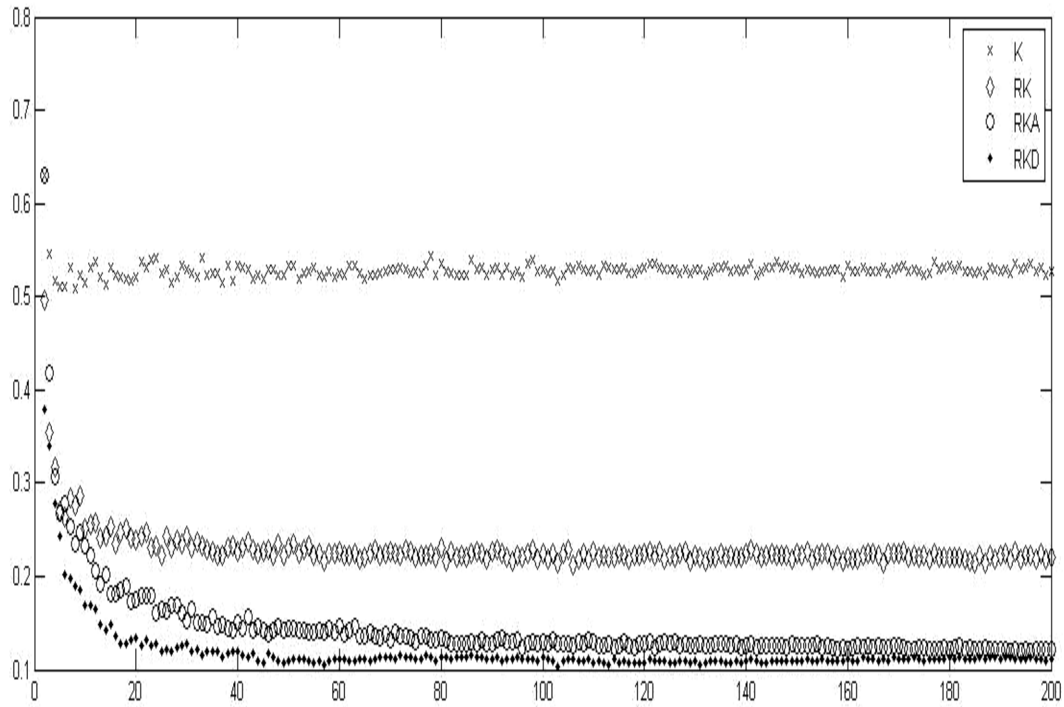

Fig. 6. Average redundancy as a function of n for each of the four versions of Kapur's code for 10^2 random $n-$tuple distributions and $n \leq 200$, $\alpha = 2$

CONCLUSION

After the investigations of the recursive versions of the Campbell code (RC) and Kapur (RK) code, we have shown that the recursive versions of both the codes are better than their original code in terms of redundancy which is considered as criteria for the comparison among the codes. Graphically, it can be easily seen that 10^4 random $n-$tuple distributions give clear picture about the recursivity as compared to 10^3 random $n-$tuple distributions and 10^2 random $n-$tuple distributions, concluding that greater the number of $n-$tuple distributions, better is the vision about recursivity. The numerical illustration provided proves that the average redundancy in case of Campbell and Kapur code lies in the vicinity of 0.5 whereas in case of their recursive versions, the average redundancy approaches towards zero. Proceeding on similar lines, the average redundancy of the other codes can be improved by the recursive approach.

ACKNOWLEDGEMENTS

The authors are thankful to Council of Scientific and Industrial Research, New Delhi, for providing the financial assistance for the preparation of the manuscript.

REFERENCES

Baer, MB. 2011. Redundancy-related bounds for generalized Huffman codes. IEEE Trans. Inf. Theory. 57: 2278-2290.

Campbell, LL. 1965. A coding theorem and Renyi's entropy. Information and Control. 8:423-429.

Chapeau-Blondeau, F., Delahaies, A. and Rousseau, D. 2011. Source coding with Tsallis entropy. Electronics Letters. 47:187-188.

Chapeau-Blondeau, F., Rousseau, D. and Delahaies, A. 2010. Renyi entropy measure of noise aided information transmission in a binary channel. Phys. Rev. E. 81:051112(1)-051112(10).

Devroye, L. 1986. Non-Uniform Random Variate Generation. Springer-Verlag.

Drmota, M. and Szpankowski, W. 2004. Precise minimax redundancy and regret, IEEE Trans. Inf. Theory. 50 (11):2686-2707.

Huffman, DA. 1952. A method for the construction of minimum redundancy codes. In: Proceedings of the IRE 40. 1098-1101.

Kapur, JN. 1998. Entropy and Coding. MSTS, New Delhi, India.

Koski, T. and Persson, LE. 1992. Some properties of generalized exponential entropies with applications to data compression. Inform. Sci. 62:103-132.

Kraft, LG. 1949. A Device for Quantizing Grouping and Coding Amplitude Modulated Pulses. MS. Thesis. Electrical Engineering Department, MIT.

Mohajer, S., Pakzad, P. and Kakhbod, A. 2012. Tight bounds on the redundancy of Huffman codes. IEEE Trans. Inf. Theory. 58 (11):6737-6746.

Narimani, H., Khosravifard, M. and Gulliver, TA. 2014. Huffman redundancy for large alphabet sources. IEEE Trans. Inf. Theory. 60(3):1412-1427.

Narimani, H., Khosravifard, M. and Gulliver, TA . 2013. How suboptimal is the Shannon code?. IEEE Trans. Inf. Theory. 59 (1):458-471.

Parkash, O. and Kakkar, P. 2014. Optimum probability distribution for minimum redundancy of source coding. Applied Mathematics. 5:96-105.

Parkash, O. and Kakkar, P. 2012. Development of two new mean codeword lengths. Inform. Sci. 207:90-97.

Ramamoorthy, A. 2011. Minimum cost distributed source coding over a network. IEEE Trans. Inf. Theory. 57:461-475.

Renyi, A. 1961. On measures of entropy and information. In: Proc. Fourth Berkeley Symp. Math. Statist. Prob. 1. 547-561.

Shannon, CE. 1948. A mathematical theory of communication. Bell Syst. Tech. J. 27:379-423.

Sharma, BD. and Raina, N. 1980. Coding theorem for partial received information. Inform. Sci. 20:181-189.

Tsallis, C. 1988. Possible generalization of Boltzmann-Gibbs statistics. J. Stat. Phys. 52:479-487.

Tu, GF., Liu, JJ., Zhang, C., Gao, SS. and Li, SD. 2011. Studies and advances on joint source-channel encoding/decoding techniques in flow media communication. Sci. China Inform. Sci. 54:1883-1894.

Ye, C. and Yeung, RW. 2002. A simple upper bound on the redundancy of Huffman codes. IEEE Trans. Inf. Theory. 48 (7):2132-2138.

Received: Sept 15, 2015; Accepted: Nov 6, 2015