A New Recursive Algorithm for Universal Coding of Integers

Mehdi Nangir*
Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran
mahdinangir@gmail.com

Hamid Behroozi
Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran
behroozi@sharif.edu

Mohammad Reza Aref
Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran
aref@sharif.edu

Received: 14/Jul/2013 Revised: 08/Sep/2014 Accepted: 13/Nov/2014

Abstract

In this paper, we aim to encode the set of all positive integers so that the codewords not only be uniquely decodable but also be an instantaneous set of binary sequences. Elias introduces three recursive algorithms for universal coding of positive integers where each codeword contains binary representation of the integer plus an attachment portion that gives some information about the first part [1]. On the other hand, Fibonacci coding which is based on Fibonacci numbers is also introduced by Apostolico and Fraenkel for coding of integers [2]. In this paper, we propose a new lossless recursive algorithm for universal coding of positive integers based on both recursive algorithms and Fibonacci coding scheme without using any knowledge about the source statistics [3].The coding schemes which don't use the source statistics is called universal coding, in these universal coding schemes we should use a universal decoding scheme in the receiver side of communication system. All of these encoding and decoding schemes assign binary streams to positive integers and conversely, without any need of use to probability masses over positive integers. We show that if we use Fibonacci coding in the first part of each codeword we can achieve shorter expected codeword length than Elias Omega code. In addition, our proposed algorithm has low complexity of encoding and decoding procedures.

Keywords: Universal Source Coding (data compression); Fibonacci Coding; Elias Coding Schemes; Integer Representation; Omega Coding; Redundancy.

1. Introduction

Researchers in the field of source coding and data compression have focused on offering efficient and fast running algorithms that have simple software and hardware implementation with expected codeword length close to entropy [4]. In this work, we aim to encode all the positive integers so that the set of codewords not only be uniquely decodable but also be an instantaneous set of binary sequences [5]. There are many situations in which the alphabet is a set of positive integers. For example, we might have a list of items and we wish to encode the position of an element in the list. Also, we may want to encode an image file by encoding the intensities.

We can classify algorithms in this field into two categories: In the first one, that we refer to as recursive algorithms, codewords have a prefix portion or a suffix portion of binary string that converts standard binary representation of positive integers to a uniquely decipherable presentation or even an instantaneous representation with minimum possible expected codeword length and low complexity encoding and decoding algorithms. In the second category, there are algorithms based on applying a complex mathematical method to compress equivalent binary representation of positive integers as much as possible. It is obvious that, with the same goal of getting shorter expected codeword length, algorithms in the second category have more complexity for encoding and decoding procedures compared with the algorithms in the first category.

Peter Elias offers three universal coding schemes to encode positive integers [1]. Universal coding means that there is no need to use the probability distribution or statistical properties of the source that we want to encode or the data that we want to compress. Elias named these three algorithms Gamma, Delta and Omega representations. Usually compared with Delta, Omega, ω, has shorter expected codeword length while Delta has shorter expected codeword length compared with Gamma. These three algorithms are in the first category; they attach a binary representation to standard binary presentation of the positive integer that provides the capability of being

* Corresponding Author

uniquely decipherable or instantaneous property to the codeword set [5]. Our proposed algorithm is also in the category of recursive schemes.

In Omega coding scheme for positive integers [1], first the binary representation of the integer is obtained. Then, we attach a prefix portion that shows the number of bits that was written in the previous step minus one in a binary format. We continue this procedure until the prefix portion be two-bit size part. At the end of the procedure, bit "0" is attached as a delimiter to indicate the end of the codeword. For example, the Omega codeword for 2012 is

$$\omega(2012) = 111010111110111000$$

Mathematically, Elias Omega codeword for any positive integer n_0 can be writtenas the following recursive structure [1], [6]

$$C_E(n_0) = [n_k]_2[n_{k-1}]_2 \dots [n_1]_2[n_0]_2 0 \qquad (1)$$

Where $[n]_2$ is the binary representation of n. Each n_k in (1) is obtained recursively by $n_k = \lfloor \log_2 n_{k-1} \rfloor$ where the recursive algorithm stops when the length of $[n_k]_2$ is two. The decoding procedure is simply the inverse of the encoding procedure: we read the first two bits of every codeword $C_E(n_0)$ to obtainn_k, and then we know that $n_k + 1$ represents the bit length of$[n_{k-1}]_2$. Thus the length of $[n_{k-1}]_2$ is recursively obtained from n_k. We continue this procedure until the first bit of the next portion is 0, and then the last portion is the number that is encoded. In fact, since the most significant bit (MSB) of every $[n_k]_2$ is "1", delimiter "0" can stop the recursion andn_0 can be easily found.

A simple coding scheme in the second category of algorithms is Fibonacci coding to encode positive integers [2], [7]. In this scheme, we write an integer number according to summation of Fibonacci numbers. The first 10 Fibonacci numbers, F_n for $n = 1,2,\cdots,10$, are shown in Table 1. To encode an integer$N$, first the largest Fibonacci number equal to or less than N is determined. If the number subtracted was the i'*th* Fibonacci number, F_i, a "1" will be placed as the i'*th* bit in the codeword. By subtracting this Fibonacci number from$N$, we repeat the previous steps for the reminder untila remainder of 0 is reached. Eventually, we add a "1" tothe rightmost digit in the codeword, which indicates the codeword is ended. The Fibonacci representation of an integer has an interesting property that it does not contain any adjacent 1's [8], so that receiving "11" string means the end of the codeword [2],[7]. For example, the Fibonacci codeword of 2012 is

$$F(2012) = 10100001000010011,$$

Since $2012 = F_1 + F_3 + F_8 + F_{13} + F_{16}$, in which F_i shows the i'*th* term in Fibonacci sequence. A disadvantage of the Fibonacci coding is that the complexity of coding and decoding algorithms increases by increasing the numbers to be coded or the length of stream to be decoded but this scheme is very efficient if larger numbers are more frequent than smaller ones [9].

Table 1: Fibonacci Numbers

$F_1 = 1, F_2 = 2,\ \ F_n = F_{n-1} + F_{n-2} \quad \forall\, n \geq 3$

F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
1	2	3	5	8	13	21	34	55	89

In this paper, we propose a new recursive algorithm to encode all of the positive integers. In this proposed coding scheme, every codeword has a prefix portion which provides the uniquely decipherable property to the codeword set. We observe that our proposed algorithm is an instantaneous set of binary codes with low complexity of encoding and decoding procedures (in both software and hardware implementations). Numerical simulations show that generally for most of probability distributions, the proposed coding has shorter expected codeword length that Elias Omega coding scheme in [1].

The paper is organized as follows. The problem definition is presented in Section 2. Our proposed universal coding for positive integers is introduced in Section 3. For comparison, the expected codeword length of our proposed algorithm is compared with that of the Elias ω code in Section 4, where two sufficient conditions on the discrete probability distribution of the input source are provided in order to get shorter expected codeword length than Omega coding. We present the decoding procedure in Section 6. Conclusions are finally drawn in Section 6.

2. Problem Definition

The general problem we encounter here is encoding the set of integer numbers without any use of probability distribution. In other words, the encoder and the decoder do not know the statistics of the input stream. This is called universal source coding or universal data compression [10]. The concept of universal coding of integers gets a lot of attention by researchers who work in source coding and data compression field [3].

Similar to [1], [6], [11]-[13] we have treated the universal coding of the positive integers that have decreasing probability distribution for positive integers, i.e.,

$$P(n) \geq P(n+1) \qquad \forall\, n \in \mathbb{N}$$

where $P(n)$ is a probability distribution on the set of positive integers, $\mathbb{N} = \{1,2,3,\cdots\}$. We assume that the inputsymbols are emitted by a memoryless information source with a possibly infinite alphabet, i.e., data emitted from the source is an independent and identically distributed (i.i.d) stream. Since we assume that the source has infinite alphabet, we can apply a one-to-one mapping between the source alphabet and the set of positive integers.

We aim to provide an encoding algorithm that encodes all the positive integers independently. Thus, the encoder is a memoryless system. Note that there are some universal encoding schemes in the literature which use memory in the encoding procedures [14]-[16].

By considering integers of the source output into blocks, each one containing n integer numbers, wecan apply our algorithm to each block separately, similar to thescheme in [17]. From the decreasing probability distribution for integers, it is clear that the codeword length would be an ascending function of$n$. Our goal is to encode positive integers with low complexity and minimum redundancy. Note that for a probability distribution P, the normalized redundancy is defined in [18] as

$$R(P) \triangleq \frac{\mathbb{E}L(P) - H(P)}{H(P)}$$

where $\mathbb{E}L(P)$ is the expected codeword length over the probability distribution P, and $H(P)$ denotes the entropy of the source. Due to first theorem of Shannon [5], we know that$R(P) \geq 0$, so we get more efficient compression if the normalized redundancy approaches to zero. Although there are some schemes that achieve an expected codeword length close to the Shannon entropy over most of probability distributions [19], [20], the drawback of these algorithms is being lossy. In this paper, we describe a lossless scheme that encodes positive integers universally with expected codeword length close to the entropy of the source due to Elias Omega codeword length [1].

3. Proposed Coding Scheme

Here, we describe our proposed algorithm. As we mentioned before, our algorithm is a recursive scheme which has three steps:

1. First, we write the standard binary representation of the positive integer number that we intend to encode and remove its most significant bit. For instance, for integer 17, we write 0001.
2. Then, we count the number of bits that we obtained in the first step (clearly, for number n we have $\lfloor \log_2 n \rfloor$ bits). For the integer number 17, we have 4 bits.
3. Finally, we attach the Fibonacci code of $\lfloor \log_2 n \rfloor$ to the left side of the binary string we obtained in the first step.

As an example, the integer number 17 has 4 bits in the first step and hence the Fibonacci codeword for integer 4 is $F(4) = 1011$. We attach 1011 to the left side of 0001 and obtain the codeword 10110001.

However, there exists one problem here. In the above algorithm, the positive integer "1" does not have any codeword. In other words, this algorithm produces the null codeword for the number "1", so we have to assign a single bit to determine the number to be encoded is "1" or not. We add a single bit at the frontier of the codewords to determine whether the number that is encoded is "1" or not. At the decoder, if the first bit is 1 we know that "1" is encoded and if the first bit is 0 we conclude that the encoded number is not "1". So that the codeword of "1" is the single bit 1, i.e., we write$C(1) = 1$. As a result, the codeword for the positive integer number 17 is $C(17) = 010110001$ where we added single "0" bit to the beginning of the codeword "10110001".

In Table 2, we provide the codewords of some positive integers based on our proposed scheme.

Table 2: Codewords of some integers based on our proposed scheme

Integer number	Our codeword
1	1
9	00011001
23	010110111
53	00001110101
78	010011001110
1000	0100011111101000
2012	00100111111011100

4. Performance Comparison between our Proposed Scheme and Omega Scheme

For comparison, the codeword lengths of the proposed scheme for some integers versus Elias ω code, which is a verystrong andfamous method in data compression and algorithmic source coding, are provided in Table 3. Based on the codeword lengths in Table 3 and also using simulation results, we observe that the only integers for which our codeword lengths exceed Elias ω codeword lengths are {2,3,8,9,10,11,12,13,14,15} (among all of the positive integers) and for the rest of integers, $\{1, 4, 5, 6, 7, 16, 17, 18, \ldots\}$, our coding scheme performs superior to Elias ω coding scheme, i.e., our codewords have shorter lengths than Elias ω codewords. Thus, our proposed algorithm results in a shorter expected codeword length over the most of discrete probability distributions, compared with the Elias ω coding scheme.

An advantage of our proposed algorithm is that if there is no integer "1" in the input sequence, we can eliminate the first bit from all of the code words, since this bit is determining whether the encoded number is "1" or not.

We apply our proposed algorithm to encode all the integers between "1 to 128". In Figure 1, we presented codeword lengths of two coding schemes: our proposed coding scheme and Elias Omega coding scheme. We observe that except for some small integers, our codeword lengths are shorter than codeword lengths of Elias ω coding scheme. More precisely, the only integers that Omega codewords have shorter lengths are 2,3,8,9,10,11,12,13,14 and 15 among all the positive integers and for the rest of integers ({1,4,5,6,7,16,17,18,...}) our codewords have shorter lengths.

Table 3: Codeword length of our proposed scheme v.s Elias ω coding scheme

Integer number	Elias ω code	Our proposed code
1	1	1
2-3	3	4
4-7	6	6
8-15	7	8
16-31	11	9
32-63	12	11
64-127	13	12
128-255	14	13

Also note that because of constructing the prefix portion of the codeword from number of bits that is written in the first step of the encoding, the codeword lengths differ for integer numbers that are different powers of 2. In other words, for every positive integer k, all of the integers from 2^{k-1} to $2^k - 1$ have equal codeword length, which can be observed in Figure 1.

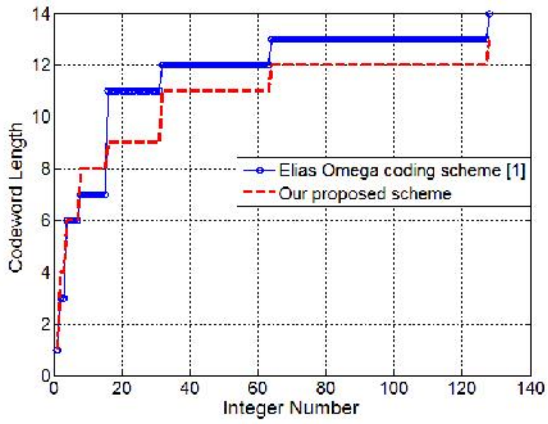

Fig. 1: Performance of our proposed algorithm compared with that of Elias Omega coding scheme.

In the next two Theorems, we provide strong sufficient conditions for discrete probability distributions over integers for which our proposed algorithm performs superior to the other algorithms such as Elias Omega coding scheme.

Theorem 1. A strong sufficient condition on the discrete probability distribution of a source to achieve shorter expected codeword length, based on our algorithm, than Omega coding scheme is

$$2(p_{16} + p_{17} + \cdots + p_{31}) \geq (p_2 + p_3) + (p_8 + p_9 + \cdots + p_{15}) \qquad (2)$$

where p_i denotes the occurrence probability of number "i" in the output sequence of the discrete source.

Proof: From the condition provided in Theorem 1, if we add the following terms to both sides of (2):

$$p_1, 3(p_2 + p_3), 6(p_4 + \cdots + p_7), 7(p_8 + \cdots + p_{15}), 9(p_{16} + \cdots + p_{31}),$$

we obtain

$$p_1 + 3(p_2 + p_3) + 6(p_4 + \cdots + p_7) + 7(p_8 + \cdots + p_{15}) + 11(p_{16} + \cdots + p_{31}) \geq p_1 + 4(p_2 + p_3) + 6(p_4 + \cdots + p_7) + 8(p_8 + \cdots + p_{15}) + 9(p_{16} + \cdots + p_{31}).$$

Let $L(\omega(n))$ and $L(C(n))$ denote the codeword lengths for the Elias Omega code and our proposed code, respectively. Since based on Table 3, we know that for the positive integers larger than "32", the codeword lengths of our proposed scheme are shorter than those of Elias ω code, we can write the following inequality

$$p_1 + 3(p_2 + p_3) + 6(p_4 + \cdots + p_7) + 7(p_8 + \cdots + p_{15}) + 11(p_{16} + \cdots + p_{31}) + \sum_{j=32}^{\infty} p_j L(\omega(j)) > p_1 + 4(p_2 + p_3) + 6(p_4 + \cdots + p_7) + 8(p_8 + \cdots + p_{15}) + 9(p_{16} + \cdots + p_{31}) + \sum_{j=32}^{\infty} p_j L(C(j)) \qquad (3)$$

If $\mathbb{E}[.]$ denotes the expectation of a random variable over the probability distribution of the source, then (3) results in

$$\mathbb{E}\big[L(\omega(n))\big] > E\big[L(C(n))\big].$$

Thus, we achieve a code whose expected codeword length is shorter than that of Elias coding scheme. This completes the proof.

For example, if we check the condition of Theorem 1, given in (2), for character probability distribution (relative frequencies of letters in the English language [21]) we observe that

$$2(p_{16} + p_{17} + \cdots + p_{31}) = 0.3895,$$

$$(p_2 + p_3) + (p_8 + p_9 + \cdots + p_{15}) = 0.379,$$

which shows validity of the condition in Theorem 1 for a practical scenario.

Theorem 2. The second strong sufficient condition on the discrete probability distribution of a source to achieve shorter expected codeword length, based on our coding scheme, than Omega coding is

$$p_{16} + p_{17} + p_{18} + \cdots \geq (p_2 + p_3) + (p_8 + p_9 + \cdots + p_{15}) \quad (4)$$

where p_i denotes the occurrence probability of number "i" in the output sequence of the discrete source.

Proof: Using (4) and adding the following terms to both sides:

$$p_1, p_2 + p_3, p_4 + \cdots + p_7, p_8 + \cdots + p_{15},$$

we get

$$1 = p_1 + p_2 + p_3 + \cdots \geq p_1 + 2(p_2 + p_3) + p_4 + \cdots + p_7 + 2(p_8 + \cdots + p_{15}) \quad (5)$$

Since we know that our codewords have one bit more than Elias ωcoding scheme only for the following positive integers: {2,3,8,9,10,11,12,13,14,15}, and for the rest of integers, our scheme has shorter codeword length compared with the Omega coding scheme, we can write (5) as

$$p_1 + 3(p_2 + p_3) + 6(p_4 + \cdots + p_7) + 7(p_8 + \cdots + p_{15}) + 11(p_{16} + \cdots + p_{31}) + \sum_{j=32}^{\infty} p_j L(\omega(j)) > p_1 + 4(p_2 + p_3) + 6(p_4 + \cdots + p_7) + 8(p_8 + \cdots + p_{15}) + 9(p_{16} + \cdots + p_{31}) + \sum_{j=32}^{\infty} p_j L(C(j))$$

which results in

$$\mathbb{E}[L(\omega(n))] > E[L(C(n))].$$

Thus, the proof is complete.

5. Decoding Procedure

Decoding scheme of a universal source code also should be universal, which means that the decoder does not know the source statistics [22].

By assuming an ideal communication channel with no error, the decoding algorithm can be described as follows:

From the first bit of the codeword, the decoder knows that the coded number is "1" or not. In other words, if the MSB of the received codeword is 1, the integer number that is encoded is "1", otherwise the integer number that is encoded is not "1". In the latter case, the decoder reads the remaining part of the codeword until it receives the first "11" substring. This last part is the Fibonacci portion of the codeword. By decoding this Fibonacci portion and assuming that it is decoded to the integer number $\log_2 n$, the decoder then reads $\log_2 n$ bits after "11" sequence. This part with adding "1" as its MSB will be the binary presentation of the number that has been encoded.

For example, if we receive the following sequence: 110101100011, by applying the decoding scheme presented here we obtain the following sequence of integers: "1-1-17-1".

6. Conclusions

In this paper, we proposed a new recursive algorithm that achieves an expected codeword length which is shorter than that of other universal coding schemes such as Elias ω coding scheme. Note that Elias ω coding scheme has the shortest expected codeword length among all three coding schemes presented in [1]. In the proposed algorithm, we applied Fibonacci coding scheme in the prefix part of the codeword. More precisely, we first obtain the standard binary representation of the positive integer number and remove its most significant bit. Then, we count the number of bits that we obtained in the first step. We then attach the Fibonacci code of the number of bits obtained in the first step to the left side of the binary string we obtained in the first step. We know that the Fibonacci representation of an integer has the interesting property that it does not contain any adjacent 1's, so that receiving "11" string means the end of prefix part. Moreover, here we provided two sufficient conditions on the discrete probability distribution of the source to get shorter expected codeword length than that of Omega coding. Eventually, we applied our proposed algorithm to some first positive integers and observed its better performance compared with the codeword length for Elias ω coding scheme.

Acknowledgments

The authors would like to thank Dr. Farzan Haddadi for his valuable comments and helpful suggestions. This work has been supported in part by the Iran NSF under Grant No. 92/32575 and by the Iran Telecommunication Research Center (ITRC).

References

[1] P. Elias, “Universal codeword sets and representations of the integers,” *IEEE Trans. Inf. Theory*, vol. 21, pp. 194–203, 1975.

[2] A. Apostolico and Aviezri S. Fraenkel, “Robust transmission of unbounded strings using Fibonacci representations,” IEEE *Trans. Inf. Theory*, vol. 33, no. 2, pp. 238 – 245, Mar. 1987.

[3] P. Andreasen, Universal Source Coding, M.Sc. thesis, Math. Dept., Univ. of Copenhagen, July 2001.

[4] Z. Chen and M. H. Lee “On Fast Hybrid Source Coding Design,” in Proc. of the 2007 International Symposium on Information Technology Convergence (ISITC 2007), pp. 143-147, Nov. 2007.

[5] T. M. Cover and J. A. Thomas, Elements of Information Theory. New York: 2nd Edition, John Wiley & Sons, 2006.

[6] T. Amemiya and H. Yamamoto, “A new class of the universal representation for the positive integers,” *IEICE Trans. Fundamentals*, vol. E76-A, no. 3, pp. 447–452, Mar. 1993.

[7] S. Fraenkel and S. T. Klein, “Robust universal complete codes for transmission and compression,” *Discrete Applied Mathematics*, vol. 64, 1996.

[8] D. E. Knuth, the Art of Computer Programming. vol. 1, 2nd Ed., Reading, MA, Addison-Wesley, 1973.

[9] S. Mohajer and A. Kakhbod, “Anti-uniform Huffman codes,” *IET Communications*, vol. 5, no. 9, pp. 1213-1219, June 2011.

[10] L. D. Davisson, “Universal noiseless coding,” *IEEE Trans. Inf. Theory*, vol. 19, no. 6, pp. 783–795, Nov. 1973.

[11] V. I. Levenshtein, “On the redundancy and delay of decodable coding of natural numbers,” *Probl. Cybern*., vol. 20, 1968.

[12] R. Ahlswede, T. S. Han, and K. Kobayashi, “Universal coding of integers and unbounded search trees,” *IEEE Trans. Inf. Theory*, vol. 43, no. 2, pp. 669–682, Mar. 1997.

[13] G. I. Shamir, “Universal Source Coding for Monotonic and Fast Decaying Monotonic Distributions,” *IEEE Trans. Inf. Theory*, vol. 59, no. 11, Nov 2013.

[14] S. Jalali, S. Verdu, and T. Weissman, “A Universal Scheme for Wyner–Ziv Coding of Discrete Sources,” *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1737-1750, April 2010.

[15] A. Beirami, M. Sardari, and F. Fekri, “Results on the fundamental gain of memory-assisted universal source coding,” in Proc. 2012 IEEE International Symposium on Information Theory (ISIT), July 2012, pp. 1092-1096.

[16] A. Beirami and F. Fekri, “Memory-Assisted Universal Source Coding,” Data Compression Conference (DCC), April 2012.

[17] D. P. Foster, R. A. Stine, and A. J. Wyner, “Universal codes for finite sequences of integers drawn from a monotone distribution,” *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1713–1720, Jun. 2002.

[18] T. S. Han and K. Kobayashi, Mathematics of Information and Coding. American Mathematical Society, 2002.

[19] S. Jalali and T. Weissman, “Near optimal lossy source coding and compression-based denoising via Markov chain Monte Carlo,” in Proc. 42nd Annual Conference on Information Sciences and Systems (CISS 2008), pp. 441-446, March 2008.

[20] S. Jalali and T. Weissman, “Lossy Source Coding via Markov Chain Monte Carlo,” in Proc. IEEE International Zurich Seminar on Communications, pp. 80-83, March 2008.

[21] D. Salomon, Data Compression- The Complete Reference. Fourth Edition, Springer, 2007.

[22] E. Ordentlich, G. Seroussi, S. Verdu, and K. Viswanathan, “Universal Algorithms for Channel Decoding of Uncompressed Sources,” *IEEE Trans. Inf. Theory*, vol. 54, no. 5, pp. 2243-2262, May 2008.

Mehdi Nangir received the B.Sc. degree in Electrical Engineering from Tabriz University and M.Sc. degree in Communication System Engineering from Sharif University of Technology in 2010 and 2012, respectively. He is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering at Khaje Nasir University of Technology, Tehran, Iran. His research interests include information theory, cryptography, data compression algorithms and source and channel coding.

Hamid Behroozi received the B.Sc. degree from University of Tehran, Tehran, Iran, the M.Sc. degree from Sharif University of Technology, Tehran, Iran, and the Ph.D. degree from Concordia University, Montreal, QC, Canada, all in Electrical Engineering in 2000, 2003, and 2007, respectively. From 2007 to 2010, he was a Postdoctoral Fellow in the Department of Mathematics and Statistics, Queen’s University, Kingston, ON, Canada. He is currently an Assistant Professor in the Electrical Engineering Department, Sharif University of Technology, Tehran, Iran. His research interests include information theory, joint source-channel coding, and cooperative communications. Dr. Behroozi is the recipient of several academic awards including Ontario Postdoctoral Fellowship awarded by the Ontario Ministry of Research and Innovation (MRI), Quebec Doctoral Research Scholarship awarded by the Government of Quebec (FQRNT), Hydro Quebec Graduate Award, and Concordia University Graduate Fellowship.

Mohammad Reza Aref was born in city of Yazd in Iran in 1951. He received his B.Sc. in 1975 from University of Tehran, his M.Sc. and Ph.D. in 1976 and 1980, respectively, from Stanford University, all in Electrical Engineering. He returned to Iran in 1980 and was actively engaged in academic and political affairs. He was a Faculty member of Isfahan University of Technology from 1982 to 1995. He has been a Professor of Electrical Engineering at Sharif University of Technology since 1995 and has published more than 260 technical papers in communication and information theory and cryptography in international journals and conferences proceedings. His current research interests include areas of communication theory, information theory and cryptography with special emphasis on network information theory and security for multiuser wireless communications. At the same time, during his academic activities, he has been involved in different political positions. First Vice President of I. R. Iran, Vice President of I. R. Iran and Head of Management and Planning Organization, Minister of ICT of I. R. Iran and Chancellor of University of Tehran, are the most recent ones.