Decoding Codes on Graphs

Low Density Parity Check Codes

A S Madhu and Aditya Nori

A S Madhu

Aditya Nori

The authors are graduate students with the Department of Computer Science and Automation, IISc. Their research addresses various aspects of algebraic and combinatorial coding theory.

The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point.

– Claude Shannon, 1948.

1. Introduction

The fifty five year old history of error correcting codes began with Claude Shannon's path-breaking paper entitled 'A Mathematical Theory of Communication' in the *Bell Systems Technical Journal* in 1948. The paper set up a well defined goal – that of achieving a performance bound set by the noisy channel coding theorem, proved in the paper. Whereas the goal appeared elusive twenty five years ago, today, there are practical codes and decoding algorithms that come close to achieving it. It is interesting to note that all known coding schemes that approach the goal can be viewed as codes on graphs with associated iterative decoding algorithms. The main ideas underlying codes on graphs were introduced by Robert Gallager in his PhD thesis written about forty years ago. Gallager's thesis was far ahead of his time and displayed remarkable prescience. However, given the limited computing power available then, Gallager's codes were not considered practical. A landmark paper by R M Tanner presented algebraic methods for constructing graphs on which efficient decoding could be implemented. A significant leap forward towards the goal set by Shannon was in the early 1990's with the discovery of turbo codes by C Berrou and A Glavieux and P Thitimajshima, who obtained excellent practical performance. However, at that time there was still

Keywords
Error correcting codes, low density parity check codes, message passing, linear time decoding, codes on graphs.

The main ideas underlying codes on graphs were introduced by Robert Gallager in his PhD thesis written about forty years ago.

no explanation for this phenomenon. A few years later D J C MacKay and R M Neal showed that Gallager type codes were competitive with turbo codes. Around the same time, M Sipser and D Spielman used graphs known as 'expander graphs' to achieve remarkable performance with moderate decoder complexity. We now know that there is a unifying view of all these codes – the representation of systems on graphs and using approximate inference algorithms for decoding. In the first part of this article we will introduce the low density parity check codes of Gallager and explain a simple algorithm presented by him for iterative decoding.

The theory of error correcting codes is concerned with the development of solutions to the following problem. We have a *sender* who wishes to transmit a *message* (a sequence of digits) to a *receiver* through a *channel* which serves as the medium of transmission. Typically this channel is not entirely reliable (that is, the channel is *noisy*) which leads to the possibility of the receiver not receiving the actual message but a corrupted version of it. Some examples of a noisy channel are:

- a telephone cable over which two modems communicate digital information which is affected by cross-talk from other lines.
- the radio communication link from a satellite to Earth with noise in the form of background radiation from terrestrial and cosmic sources.
- a disk drive where defects may cause the head to report wrong values for binary digits.

M Sipser and D Spielman used graphs known as 'expander graphs' to achieve remarkable performance with moderate decoder complexity.

Therefore the question to ask is: "Is it possible to ensure reliable transmission inspite of errors introduced by the noisy channel?" This problem was studied by Shannon and led to the notion of *channel coding* where the message bits sent by the receiver are 'padded' with

Raj Chandra Bose (1901-87)

The February 2002 issue of *Resonance* was dedicated to Claude Shannon, referred by many as the 'father of information theory' and many consider the appearance of his paper 'A mathematical theory of communication' in 1948 as heralding the beginning of the Information Age (see the Article-In-A-Box by Priti Shankar in that issue). To a good measure both these attributes apply to Raj Chandra Bose (1901-1987) often referred to as the 'father of experimental design.' Shannon's theorem on the possibility of information transmission over a noisy channel with as low an error as desired was an 'existential' result but not a 'constructive' one. The construction of such a code evolved from the work of Raj Chandra Bose culminating in the construction of Bose–Chaudhuri–Hocquenghem (BCH) error correcting codes in 1960. Variations of these codes are the ones in wide use today for all modes of digital information transmission. Bose used to describe these codes as "a technique which will make errors in transmission of information so infrequent that it will be surprising if there was one error in hundred years of transmissional communication." It is interesting to note that Joseph George Caldwell has made the following comment – "It is obvious why Bose was never awarded a Nobel Prize (for the BCH codes, for solving Euler's conjecture, or as father of the mathematical basis for experimental design) since he was a mathematician."

Raj Chandra Bose was born on June 29, 1901 in India and had his school education in Delhi. After obtaining a master's degree in applied mathematics from the University of Delhi, Bose moved to Calcutta and got his master's degree in pure mathematics from the Calcutta University. It is interesting that he got master's degree in applied as well as pure mathematics and is indicative of the fact that his later research work encompassed both these aspects of mathematics. His first job was at Ashutosh College, an Undergraduate college in Calcutta. Here he started working on geometry and produced several papers on hyperbolic geometry. He shifted to the Department of Pure Mathematics at Calcutta University a couple of years later.

In 1932-33 P C Mahalnobis, founder of the Indian Statistical Institute (ISI) (See *Resonance* Vol.4, No.6), was able to get Bose interested in statistical problems. Bose's papers in statistics started appearing from the very first issue of *Sankhya, Indian Journal of Statistics*. Soon he was making important contributions to statistics along with inspiring and guiding many students. In 1949 he moved to University of North Carolina at Chapel Hill, USA. He built a strong school of statistics there which is flourishing even today. In 1971 he accepted an offer from Colorado State University at Fort Collins where he remained till the end.

Raj Chandra Bose made many significant contributions to several topics in mathematics and statistics. The proof of falsity of a conjecture of Euler about the non-existence of two mutually orthogonal latin squares of order 2 modulo 4 by Bose and his co-workers, Parker and Shrikhande made it to the front page of the Sunday Edition of the *New York Times* of April 26, 1959! This result earned them the nickname 'Euler Spoilers.'

He was an inspiring teacher and many of his students went on to make remarkable contributions to mathematics and statistics. He had a flair for languages and could recite verses in Arabic, Bengali, Persian, Sanskrit and Urdu. One of his friends said of Bose "... he was a great conversationalist in spite of the fact that he would hardly allow anybody else to speak!"

PS: In spite of his extraordinary achievements, biographical details of R C Bose's life are hard to locate. I would be grateful for any references from readers.

C S Yogananda
Indian Institute of Science, Bangalore

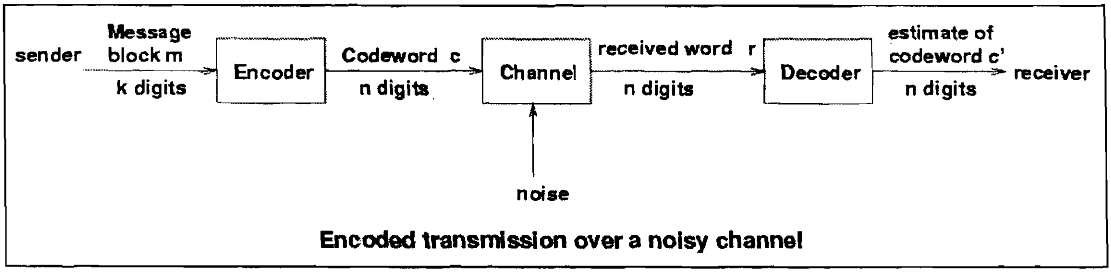


Figure 1. A typical error control scheme.

redundant *check bits* so as to protect them from errors introduced by the channel. This process of coding is shown in *Figure* 1 where the original message is transformed (encoded) to a codeword by an *encoder* and this codeword is sent through the channel. At the receiving end of the channel there is a decoder which remaps (decodes) the channel output back to a message that is read by the receiver. The set of all transformed messages forms the code. The art of error control coding involves the design of encoders and decoders that increase the reliability of transmission over noisy channels while ensuring that the amount of redundancy added to messages is not too large. Shannon proved the remarkable result that there exist codes for which the decoder can correct an arbitrary number of errors with high probability as long as the amount of redundancy in the codeword is greater than a certain value, which is now called the *Shannon limit* of the channel.

Shannon proved the remarkable result that there exist codes for which the decoder can correct an arbitrary number of errors with high probability as long as the amount of redundancy in the codeword is greater than a certain value, which is now called the *Shannon limit* of the channel.

Among the earliest discovered codes that approach the Shannon limit were the *low density parity check* (LDPC) codes. The term low density arises from the property of the *parity check matrix* defining the code. We will now define this matrix and the role that it plays in decoding.

2. Linear Codes

The parity check matrix is one way of defining a linear block code. Linear block codes are a very important class of codes in the algebraic theory of coding. The symbols that are transmitted over the channel are from a finite field. An encoder for a block code is a function for converting a sequence of message digits $\mathbf{u}$, of length k, into a transmitted sequence $\mathbf{c}$ of length n called a codeword, where n is greater than k. In an (n, k) linear block code C, the extra $n - k$ digits are linear functions of the original k digits and these are called *parity check digits*. Apart from n and k, another important parameter for a code is d, the *minimum distance* of the code. The distance between two codewords (also called the *Hamming distance*) is defined to be the number of positions in which these codewords differ. The *minimum distance* of a code is the minimum of distances over all pairs of codewords. For a linear code the minimum distance turns out to be the minimum number of non-zero components in any codeword. The minimum distance of a code plays an important role in its error correcting ability. It is easily shown that a code of minimum distance d can correct up to $\lfloor \frac{d-1}{2} \rfloor$ errors where a single error is a digit of the transmitted codeword that is erroneously received. An (n, k) linear code can be represented compactly by a $k \times n$ matrix as follows. The n digit transmitted sequence $\mathbf{c}$ can be obtained from the k digit message sequence $\mathbf{u}$ by a linear operation,

$$\mathbf{c} = \mathbf{u}G, \tag{1}$$

where G is the *generator matrix* of the code and the encoding operation in (1) uses modulo-2 arithmetic for a binary code $(0 + 0 = 0,\ 0 + 1 = 1,\ 1 + 0 = 1,\ 1 + 1 = 0)$. The rows of the generator matrix are linearly independent over the field over which the code is defined. The generator matrix for the (4, 2) linear binary block code with codewords $\{(0000), (0110), (1001), (1111)\}$ is shown in *Figure* 2.

Figure 2. A generator matrix for (4,2) linear block code.

$$G = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

An alternate way of specifying the code C is by an $(n - k) \times n$ parity check matrix H of C which enjoys the following property:

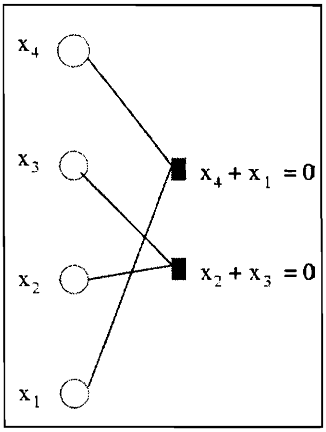
$$H\mathbf{c}^T = \mathbf{0},\ \forall \mathbf{c} \in C.$$

Thus whereas the generator matrix defines the vector subspace which is the code, the parity check matrix defines the orthogonal subspace. For the $(4,2)$ code that we considered earlier, the parity check matrix is the same as the generator matrix G shown in *Figure* 2. Such codes are referred to as *self dual* codes. Each row of the parity check matrix can be thought of as a constraint that the digits of the codeword must satisfy. Thus a codeword is a vector whose digits simultaneously satisfy all the $n-k$ constraints imposed by the linearly independent rows of the parity check matrix.

Given a received sequence **y**, the *problem of decoding* over a graph $\mathcal{B}$ is essentially that of efficiently finding a sequence (with minimal distance from **y**) whose components simultaneously satisfy all the constraints in $\mathcal{B}$.

This constraint system of equations defined by the parity check matrix may be depicted pictorially by way of a bipartite graph, where nodes are partitioned into variable nodes which represent codeword components, and constraint nodes that represent parity check constraints. There is an edge (u,v) from a variable node u to a constraint node v if and only if the variable u participates in the constraint v. For example, the $(4,2)$ code would have four variables (call them x_1, x_2, x_3 and x_4) and two constraints which are $x_2 + x_3 = 0$ and $x_1 + x_4 = 0$. The bipartite graph $\mathcal{B}$ for this code is shown in *Figure* 3. Given a received sequence $\mathbf{y}$, the *problem of decoding* over a graph $\mathcal{B}$ is essentially that of efficiently finding a sequence (with minimal distance from $\mathbf{y}$) whose components simultaneously satisfy all the constraints in $\mathcal{B}$.

Figure 3. The graph representing the (4,2) linear block code.



3. Solving Linear Equations

Let us now digress from the problem of codes and decoding, and turn our attention to solving linear systems of equations. Suppose we were required to solve a system of m linear equations in n unknowns. The standard method of solving them would be to use the ever popular Gaussian elimination method. But this would require $O(n^3)$ (that is, time cubic in the size of the input) computations. However, if we were given the values of a subset of the variables and were required to find satisfying values for the remaining variables, we might be able to carry out the task more efficiently as the following example illustrates. Consider the following linear system of 3 equations in 5 unknowns and assume that all arithmetic is performed modulo 2.

$$\begin{aligned} x_1 + x_2 &= 0 && (2)\\ x_3 + x_4 + x_5 &= 0 && (3)\\ x_1 + x_4 + x_5 &= 0. && (4) \end{aligned}$$

Suppose we know that $x_1 = 1$ and $x_4 = 1$. Substituting for these variables in the above equations, we have

$$\begin{aligned} 1 + x_2 &= 0 && (5)\\ x_3 + 1 + x_5 &= 0 && (6)\\ 1 + 1 + x_5 &= 0. && (7) \end{aligned}$$

Shifting known values to the right we obtain

$$\begin{aligned} x_2 &= 1 && (8)\\ x_3 + x_5 &= 1 && (9)\\ x_5 &= 0. && (10) \end{aligned}$$

From (8) and (10), $x_2 = 1$ and $x_5 = 0$ and again substituting these values in the equations results in

$$x_3 = 1. \qquad (11)$$

Thus we have obtained the remaining values using no linear algebra. It might appear that we were just lucky and cases illustrated by the example above are extremely rare. However it turns out that in the context of coding we can ensure by a proper choice of the code that this desirable phenomenon happens with high probability. In fact, this idea of 'iteratively' solving linear equations is fundamental to the decoding of LDPC codes. In order to explain the decoding procedure for LDPC codes, we

Suggested Reading

[1] Robert G Gallager, *Low Density Parity Check Codes*, PhD thesis, MIT, 1963.

[2] R Michael Tanner, A recursive approach to low complexity codes, *IEEE Trans. Inform. Theory*, Vol. IT-27, No. 5, pp.533-547, September, 1981.

[3] C Berrou, A Glavieux and P Thitimajshima, Near Shannon limit error-correcting coding and decoding: Turbo-Codes(1), *Proc. of IEEE ICC'93*, Geneva, pp. 1064-1070, June 1993.

[4] M Sipser and D A Spielman, Ex-pander codes, *IEEE Trans. Information Theory*, Vol. 42, No. 6, pp. 1710-1722, Nov 1996 .

[5] C E Shannon, A mathematical theory of communication, *Bell System Technical Journal*, Vol. 27, pp. 379-423; 623-656, July and October, 1948.

[6] D J C McKay and R M Neal, Near Shannon limit performance of low-density parity check codes, Electronics Letters, Vol.33, No.6, pp.457-458, Mar 1997.

[7] M Luby, M Mitzenmacher, M A Shokrollahi and D A Spielman, Efficient Erasure Correcting Codes, *IEEE Trans. Inform. Theory*, Vol.47, No.2, pp.569-584, February 2001.

The idea of iterativity solving linear equations is fundamental to the decoding of LDPC codes.

will also be required to fix the notion of a channel model. The channel model that we will consider is the Binary Erasure Channel (BEC) which is the simplest model of a communication channel. The input to the channel are binary digits $\{0, 1\}$ and given an input, say a, the output of this channel is either a (remains unchanged) or? (is erased). We are now in a position to explain the decoding procedure for an LDPC code. We will motivate the general idea by closely examining the linear system of equations that we considered earlier in this section. The equations 2, 3 and 4 can written in matrix form as

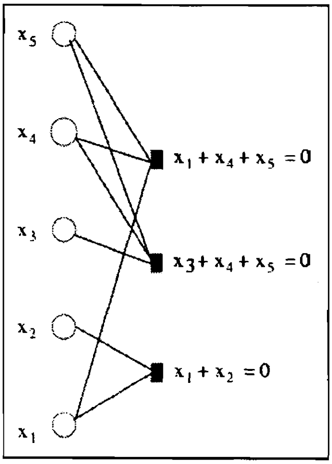
$$H = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

and this forms the parity check matrix of our code. The graph for this code is shown in *Figure* 4. Let $\mathbf{y} = (1, ?, ?, 1, ?)$ be the received word or the output of the transmitted codeword over a BEC. Since $\mathbf{y}$ is a codeword it must satisfy the equation $H\mathbf{y}^T = \mathbf{0}$. Denoting the erased digits at positions 2, 3 and 5 by x_2, x_3 and x_5 we have

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix} (1, x_2, x_3, 1, x_5)^T = \mathbf{0}.$$

Therefore the problem of decoding has been reduced to that of solving the above system of equations, and our earlier discussions show how this task may be accomplished iteratively.

Figure 4. The graph for the code represented by H.



Thus, from the solution $(x_2, x_3, x_5) = (1, 1, 0)$, the decoder outputs $(1, 1, 1, 1, 0)$ as its estimate of the transmitted codeword. Therefore the problem of decoding over the BEC may be cast as the problem of solving linear systems of equations with unknowns for erased positions. This system of equations is guaranteed to have at least one solution as the transmitted codeword satisfies these equations. But how efficient is this procedure? In order to answer this, we·model the decoding procedure as a message passing scheme between the left and right partitions of the bipartite graph representing the code. This is the topic of the next section.

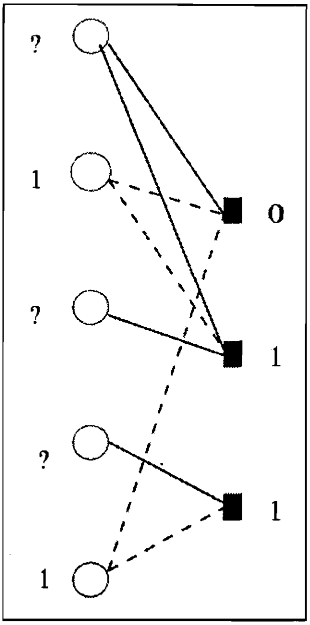
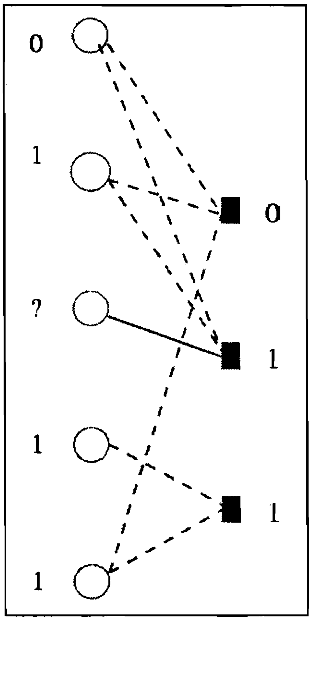


Figure 5. The graph obtained after first iteration of the MP-Decoder.

Figure 6. The graph obtained after second iteration of the MP-Decoder.



4. A Simple Message Passing Decoder

We will now describe the Message Passing decoder (MP-Decoder) which essentially mimics the procedure given in Section 3 for solving linear equations and makes use of the graph given in *Figure* 4 for this purpose. Recall that the nodes on the left represent variables and those on the right represent constraints. Initially all the variables are associated with the received sequence components and each constraint node is associated with a value 0. Each left node that is associated with a non-erasure (that is, only variables not having value?) propagates its value along all its edges. At the right, each constraint node computes and stores the modulo-2 sum of its local value with the values received along its edges. These edges are then disassociated from the graph (shown in *Figure* 5 by dashed edges).

Then each check node of degree one sends back the computed value to its neighbour which takes this value. This is shown in *Figure* 6 and the iterations are repeated till the values of all left variable nodes are known as shown in *Figure* 7. As every iteration involves the deletion of at least one edge, the decoding complexity will be $O(e)$ steps where e is the number of edges in the bipartite graph representing the code. In case of LDPC codes, which have sparse parity check matrices, the edge cardinality of the graph representing the code is linear in the number of nodes, thereby implying that the MP-Decoder for these codes has linear time decoding complexity. Gallager also analysed the probability of decoding error, that is, the likelihood that the MP-Decoder fails to output the correct answer. This was done in the context of a Binary Symmetric Channel (BSC) model. The input and output alphabets for this model consist of binary digits ({0, 1}), and the effect of noise on a transmitted bit causes it to flip[1]. Let us now see how an MP-Decoder could be designed for this channel model. Given a received sequence, the MP-Decoder computes all the constraint equations. If all of them are satisfied, then the received sequence is a codeword and the decoder exits successfully, else, the decoder flips all variables which participate in more unsatisfied than satisfied equations and the whole process is repeated with these new values. As is the case with LDPC codes, the constraint equations will consist of a small constant fraction of variables. Therefore it is reasonable to assume that toggling a variable that occurs in more unsatisfied than satisfied equations will result in an increase in the total number of satisfied equations.

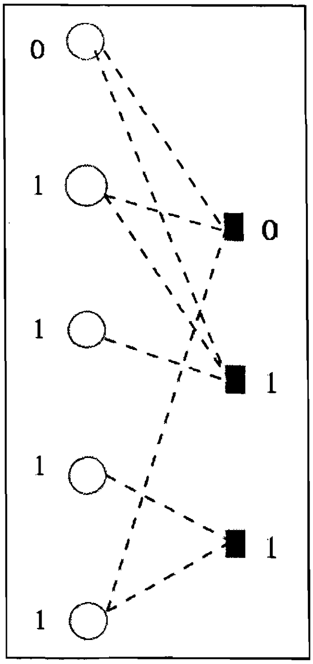


Figure 7. The final stage of the MP-Decoder.

To understand this better, let us now examine a special case. Consider a variable x_2 whose parity check set contains more than one error. Assume for the moment, that the bipartite graph representing the code can be unrolled to get a tree rooted at x_2 as shown in *Figure* 8 (Note that this need not always be true, as the bipartite graph might contain cycles).

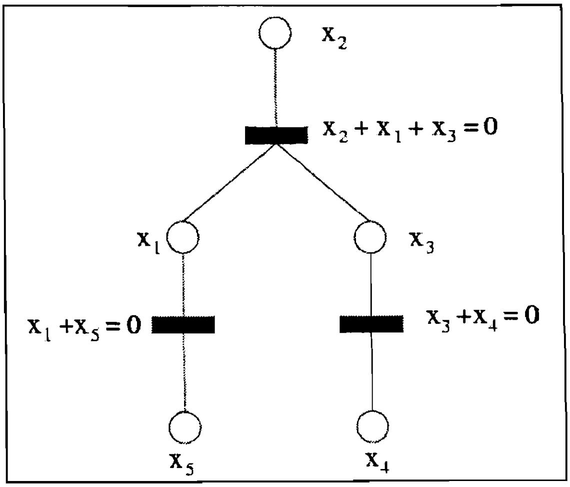


Figure 8. Unrolling a bipartite graph representing the code.

[1] In contrast to the BEC, the BSC model does not accommodate erasures.

The odd levels of the tree represent the variables while the even levels represent the constraint equations. Since x_2's parity-check set contain more than one error many variables in the third level of the tree are also errors. But assuming that variables lower down the tree contain fewer errors, the error-free variables and their constraint equations can help in correcting the variables higher up the tree. This is propagated up the tree with each iteration of the MP-decoder and thus finally x_2 gets corrected.

Recently M Luby, M Mitzenmacher, M A Shokrollahi and D A Spielman have used ideas from low-density parity check codes to construct encoding and decoding schemes for correction of erasures representing packet losses on networks. The codes are derived from cascades of sparse bipartite graphs using novel ideas for construction of the graphs in each of the layers. A message consisting of 640000 packets is encoded into a vector of 1280000 packets, and each packet consists of 256 bytes. (i.e. each message symbol is represented by 256 bytes). Luby and others [7] were able to obtain throughputs of roughly 280Mbit/s on a DEC-alpha machine with 300MHz and a 64-Mbyte RAM, and reach rates just below channel capacity, or in other words, come very close to the goal set by Shannon. It is interesting that the seeds of these ideas were sown forty years ago!

Luby and others have constructed codes that come very close to the goal set by Shannon!

In the second part of this article we will carry out a formal analysis of the Gallager algorithm and introduce a probabilistic version.

Address for Correspondence
A S Madhu and Aditya Nori
Department of Computer
Science and Automation
Indian Institute of Science
Bangalore 560012, India.
Email: madhu@csa.iisc.ernet.in
aditya@csa.iisc.ernet.in

We know very little, and yet it is astonishing that we know so much, and still more astonishing that so little knowledge can give us so much power.

Bertrand Russell (1872-1970)
English philosopher, mathematician