2004 International Symposium on Nonlinear Theory and its Applications (NOLTA2004)
Fukuoka, Japan, Nov. 29 - Dec. 3, 2004

New Directions in Coding: From Statistical Physics to Quasigroup String Transformations

Danilo Gligoroski[†], Smile Markovski[†], and Ljupco Kocarev[‡]

[†]Institute of Informatics
Faculty of Natural Sciences – Skopje
P. O. Box 162, Macedonia
E-mail: danilo@pmf.ukim.edu.mk, smile@ii.edu.mk
[‡]Institute for Nonlinear Science
University of California San Diego
9500 Gilman Drive, La Jolla, CA 92093–0402, USA
Email: lkocarev@ucsd.edu

Abstract—The last several years have witnessed major theoretical advances in communications and coding theory resulting in a new coding concept called codes on graphs. Codes on graphs, and specially their prime examples, low-density parity check (LDPC) codes and turbo codes is a research area of great current interest. In this paper we first overview the basic relation between codes on graph and spin systems, iterative decoding algorithms and nonlinear dynamical systems, and power-law networks and codes on graphs. Then, we show that there exists a class of codes, generated by quasigroup string transformations, which are almost random and outperform significantly Turbo and LDPC codes. For example, for $SNR = -0.5$ dB, rate $1/4$ and block length of only 864 bits produce $BER = 4.1 \times 10^{-5}$.

1. Introduction

The non-constructive proof of the noisy-channel coding theorem shows that good block codes exist for any noisy channel, and moreover that nearly all block codes are good. However, writing down an explicit and practical encoder and decoder that are as good as proved by Shannon in his seminal work *A Mathematical Theory of Communication* [1] is still an unsolved problem.

Recently, it has been recognized that two classes of codes, namely *turbo codes* and *low-density parity-check* (LDPC) codes, perform at rates extremely close to the Shannon limit. Turbo and LDPC codes are based on a similar philosophy: constrained random code ensembles, described by some fixed parameters plus randomness, decoded using iterative algorithms or message passing decoders.

The relationship between codes on graphs and spin models was discovered and described in a series of papers [2, 3, 4]. Following this relation, it has been shown that turbo codes correspond to coupled spin chains, while LDPC represent spin models on diluted graphs. Moreover, we suggested a link between coding theory (iterative algorithms) and chaos theory [5]. Furthermore, recently it was shown that the world's best performer LDPC codes and the Tornado codes have power law degree distributions [6].

In this paper we first briefly overview the basic relation between codes on graph and spin systems, iterative decoding algorithms and nonlinear dynamical systems, and power-law networks and codes on graphs. Then we describe a class of error correcting codes with the following property: for an arbitrary codeword C, the distribution of substrings of C of length r is uniform. An instance of such codes implemented with quasigroup string transformations is described in detail. Our preliminary numerical simulations show that proposed codes outperforms significantly corresponding turbo and LDPC codes.

2. Coding Theory and Statistical Physics

It is known that error correcting codes can be mapped onto disordered spin models [2, 3, 4]. Equivalent spin models have been intensively studied in the last few years. These are diluted spin glasses, i.e., spin glasses on random (hyper)graphs. The new codes are decoded by using approximate iterative algorithms, which are closely related to the cavity approach to mean-field spin glasses. For example, one can define two different performance measures for evaluating LDPC codes. The first is the practical performance achievable in feasible time scales that grow polynomially with the systems size; while the other is the optimal theoretically achievable performance, for which the required computation typically increases exponentially with respect to the message length. Utilizing the similarity between LDPC codes and Ising spin systems, statistical physics provides a scheme for evaluating both performance measures within the same framework; the current standard method used in the information theory community can only provide an estimate of the practical performance, and practically reduces to the one used within the statistical physics framework.

Recently, Kim and Ko [6] have investigated the potential of scale-free networks as error-correcting codes. They found that irregular low-density parity-check codes with the highest performance known to date have degree distributions well fitted by a power-law function $p(k) \sim k^{-\gamma}$ with γ close to 2, which suggests that codes built on scale-free networks with appropriate power exponents can be good error-correcting codes, with a performance possibly approaching the Shannon limit. They demonstrated for an erasure channel that codes with a power-law degree distribution of the form $p(k) = C(k + \alpha)^{-\gamma}$, with $k \geq 2$ and suitable selection of the parameters α and γ, indeed have very good error-correction capabilities.

It is known [7, 5] that iterative decoding algorithms may be viewed as complex nonlinear dynamical systems. Richardson [7] has developed a geometrical interpretation of the turbo-decoding algorithm, and formalized it as a discrete-time dynamical system defined on a continuous set. In the formalism of [7], the turbo-decoding algorithm appears as an iterative algorithm aimed at solving a system of $2n$ equations in $2n$ unknowns, where n is the block-length of the turbo code at hand. If the algorithm converges to a codeword, then this codeword constitutes a solution to this system of equations. Recently, in [5] we studied iterative decoding algorithms as discrete-time nonlinear dynamical systems. We proposed a simplified representation of several well-known iterative decoding schemes in terms of *a posteriori* average entropy. We found that, in general, iterative decoding algorithms may exhibit a whole range of phenomena known to occur in nonlinear systems. These phenomena include the existence of multiple fixed points, oscillatory behavior, and even chaos (for a finite block-length). We have also shown how the general principles distilled from our analysis may be applied to enhance the performance of existing iterative decoding schemes.

3. Almost Random Codes Based on Quasigroup String Transformations

3.1. Description of the Code

A source of information produces a stream of symbols. The stream is partitioned into blocks of length N_{block}. Each of the possible $2^{N_{block}}$ blocks is mapped to a *codeword* (i.e., a sequence of bits) of length $N > N_{block}$ by the encoder and transmitted through the channel. Therefore, an error correcting code is defined as a mapping $T : \{0,1\}^{N_{block}} \to \{0,1\}^N$.

The code T proposed in this paper is defined as follows. Let M be a block of N_{block} bits. First we add zero bits and produce a block L of length N. Second, we rewrite L as $L = L_1 L_2 \dots L_p$, where each L_i is a block of s bits (we assume that $N = sp$). Third, the block L is mapped with a bijection to a block $C = C_1 C_2 \dots C_p$ in the following way.

Let $k_{1,1}, k_{1,2}, \dots, k_{1,n}$ be n initial strings each of length s. The block L is mapped to C as

```
For $i =$  $1, 2, \dots p$
           $b_{i,0} = L_i$
For $j =$  $1, 2, \dots n$
           $b_{i,j} = f(k_{i,j}, b_{i,j-1})$
           $k_{i+1,j} = g(k_{i,j}, b_{i,j-1})$
End $j$
           $C_i = b_{i,n}$
End $i$                                          (1)
```

where f and g are appropriate operations. Note that (1) defines uniquely our code T.

If we write $k^{(i)} = k_{i,1} k_{i,2} \dots k_{i,n}$, $i = 1, 2, \dots, p+1$, then equation (1) defines also the following two maps. First, a map $F : A^n \times A^p \to A^n \times A^p$ such that $(k^{(p+1)}, C) = F(k^{(1)}, L)$, where $A = \{0,1\}^l$ is a set of all strings with length l. Second, a map $G_1 : A^n \times A^1 \to A^n \times A^1$ such that $(k^{(i+1)}, C_i) = G_1(k^{(i)}, L_i)$. For this reason we say that our code is *double iterative*: (i) for each L_i, f is iterated n times to produce C_i; and (ii) $k^{(1)}$ is iterated p times to give $k^{(p+1)}$.

In the following instead of G_1 we will work with the map $G_4 \equiv G$. For simplicity only let assume that $p = 4r$. Then equation (1) can be rewritten as

```
For $l =$  $1, 2, \dots r$
           $L^{(l)} = L_{4l-3} L_{4l-2} L_{4l-1} L_{4l}$
For $i =$  $4l-3, 4l-2, 4l-1, 4l$
           $b_{i,0} = L_i$
For $j =$  $1, 2, \dots n$
           $b_{i,j} = f(k_{i,j}, b_{i,j-1})$
           $k_{i+1,j} = g(k_{i,j}, b_{i,j-1})$
End $j$
           $C_i = b_{i,n}$
End $i$
           $C^{(l)} = C_{4l-3} C_{4l-2} C_{4l-1} C_{4l}$
End $l$.                                         (2)
```

Equation (2) defines a map $G : A^n \times A^4 \to A^n \times A^4$ such that $(k^{(4l+1)}, C^{(l)}) = G(k^{(4l-3)}, L^{(l)})$.

In addition, our code has the following property, which will be proven in the next section: the code is *almost random*, which means that for every $M \in \{0,1\}^{N_{block}}$, the distribution of substrings of $C = T(M) \in \{0,1\}^N$ of length k, $1 \leq k \leq n$, when N is large enough, is uniform.

3.2. Quasigroup String Transformations

We give a brief overview of quasigroup operations and quasigroup string transformations (more detail explanation the reader can find in [8], [9]).

A quasigroup is a groupoid $(Q, *)$ satisfying the law

$$(\forall u, v \in Q)(\exists! x, y \in Q)(u * x = v \; \& \; y * u = v).$$

This implies the cancellation laws $x * y = x * z \implies y = z$, $y * x = z * x \implies y = z$ and the equations $a * x = b$, $y * a = b$ have unique solutions x, y for each $a, b \in Q$. If $(Q, *)$ is a quasigroup, then $*$ is called a quasigroup operation.

Consider an alphabet (i.e. a finite set) A, and denote by A^+ the set of all nonempty words (i.e. finite strings) formed by the elements of A. The elements of A^+ will be denoted by $a_1a_2\dots a_n$ rather than $(a_1, a_2, \dots, a_n)$, where $a_i \in A$. Let $*$ be a quasigroup operation on the set A. For each $l \in A$ we define a function $e_{l,*} : A^+ \to A^+$ as follows. Let $a_i \in A$, $\alpha = a_1a_2\dots a_n$. Then

$$e_{l,*}(\alpha) = b_1\dots b_n \iff b_{i+1} = b_i * a_{i+1} \tag{3}$$

for each $i = 0, 1, \dots, n-1$, where $b_0 = l$. The function $e_{l,*}$ is called an e-transformation of A^+ based on the operation $*$ with leader l.

Several quasigroup operations can be defined on the set A and let $*_1, *_2, \dots, *_k$ be a sequence of (not necessarily distinct) such operations. We choose also leaders $l_1, l_2, \dots, l_k \in A$ (not necessarily distinct either), and then the composition of mappings

$$E_k = e_{l_1,*_1} \circ e_{l_2,*_2} \circ \dots \circ e_{l_k,*_k}$$

is said to be an E-transformation of A^+. The function E_k which is actually a permutation, have many interesting properties, and for our purposes the most important one is the following:

Theorem 1 ([9]) *Consider an arbitrary string $\alpha = a_1a_2\dots a_n \in A^+$, where $a_i \in A$, and let $\beta = E_k(\alpha)$. If n is large enough integer then, for each $l : 1 \le l \le k$, the distribution of substrings of β of length l is uniform. (We note that for $l > k$ the distribution of substrings of β of length l may not be uniform.)*

3.3. Software Implementation

The coding has two parts. We now describe a design of a 1/2 rate code only; the generalization to different coding rates is straightforward. Suppose that the message to be sent has the form $M = m_1m_2\dots m_{18}$, where m_i are nibbles (4-bit letters). In the first part, we add redundant information and obtain $L = L^{(1)}L^{(2)}L^{(3)}L^{(4)}L^{(5)}_5L^{(6)}L^{(7)}L^{(8)}L^{(9)}$, where $L^{(1)} = m_1m_2m_30_4$, $L^{(2)} = m_4m_5m_60_4$, $L^{(3)} = m_7m_8m_90_4$, $L^{(4)} = 0_40_40_40_4$, $L^{(5)} = m_{10}m_{11}m_{12}0_4$, $L^{(6)} = m_{13}m_{14}m_{15}0_4$, $L^{(7)} = m_{16}m_{17}m_{18}0_4$, $L^{(8)} = 0_40_40_40_4$, $L^{(9)} = 0_40_40_40_4$, where 0_4 is the string of 4 zeros (zero nibble). Therefore, each $L^{(i)}$ is a string of 16 bits. Since we add 18 zero nibbles, the rate of the code is 1/2. This is schematically shown on Table 1; in this table we also show rate 1/4 code. For this 1/2 code we also say that it is (72,144) code (the length of M is 72, the length of L is 144).

In the second part of the coding we choose f to be the quasigroup operation defined in Table 2 and g to be

$$\begin{aligned} k_{i+1,j} &= b_{i,j} \text{ if } j = 1, \dots n-1 \\ k_{i+1,n} &= b_{i,1} \oplus \dots \oplus b_{i,n}. \end{aligned}$$

The left parastrophe $Qpar(\backslash)$ of the quasigroup $Q(*)$ that we used in our experiments for the decoding algorithm is shown in Table 3.

Table 1: Codes with rates 1/2 and 1/4

Rate 1/2				Rate 1/4			
m_1	m_2	m_3	0	m_1	m_2	0	0
m_4	m_5	m_6	0	m_3	m_4	0	0
m_7	m_8	m_9	0	m_5	0	0	0
0	0	0	0	0	0	0	0
m_{10}	m_{11}	m_{12}	0	m_6	m_7	0	0
m_{13}	m_{14}	m_{15}	0	m_8	0	0	0
m_{16}	m_{17}	m_{18}	0	m_9	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Table 2: A quasigroup of order 16

*	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	3	c	2	5	f	7	6	1	0	b	d	e	8	4	9	a
1	0	3	9	d	8	1	7	b	6	5	2	a	c	f	e	4
2	1	0	e	c	4	5	f	9	d	3	6	7	a	8	b	2
3	6	b	f	1	9	4	e	a	3	7	8	0	2	c	d	5
4	4	5	0	7	6	b	9	3	f	2	a	8	d	e	c	1
5	f	a	1	0	e	2	4	c	7	d	3	b	5	9	8	6
6	2	f	a	3	c	8	d	0	b	e	9	4	6	1	5	7
7	e	9	c	a	1	d	8	6	5	f	b	2	4	0	7	3
8	c	7	6	2	a	f	b	5	1	0	4	9	e	d	3	8
9	b	e	4	9	d	3	1	f	8	c	5	6	7	a	2	0
a	9	4	d	8	0	6	5	7	e	1	f	3	b	2	a	c
b	7	8	5	e	2	a	3	4	c	6	0	d	f	b	1	9
c	5	2	b	6	7	9	0	e	a	8	c	f	1	3	4	d
d	a	6	8	4	3	e	c	d	2	9	1	5	0	7	f	b
e	d	1	3	f	b	0	2	8	4	a	7	c	9	5	6	e
f	8	d	7	b	5	c	a	2	9	4	e	1	3	6	0	f

3.4. Results

The transmitted code word is C. Due to the noise, a different sequence of symbols $D = d_1d_2\dots d_{4r}$, where d_i is a nibble received. The decoding problem is to infer L, given D, the definition of the code, and the properties of the noisy channel. Since the operations in our algorithm are discrete ones, it is applicable straightforward on a binary symmetric channel, however, if the channel is continuous additive (like AWGN), the additional soft information obtained for every received bit can be used for decreasing the number of decoding candidates in sets S_i (see below).

In the process of decoding, we iteratively decode 4-tuples $D^{(i)} = d_jd_{j+1}d_{j+2}d_{j+3}$, $j = 1 + 4(i-1)$, $i = 1, 2, \dots r$, and check if 0_4 is a nibble of the corresponding $L^{(i)}$, or if $L^{(i)}$ is a string of zeros only. However, since $D^{(i)} = d_jd_{j+1}d_{j+2}d_{j+3}$, $j = 1 + 4(i-1)$, $i = 1, 2, \dots r$, differs from the corresponding codeword $C^{(i)} = c_jc_{j+1}c_{j+2}c_{j+3}$ in some bits, in process of decoding we decode every 4-tuple which is less than B bits distant from $D^{(i)}$. In a few words: decoding of the codeword is a process of a search of those $D^{(i)}$s for which, when decoded, the last nibble is a string of 4 zeros, and $L^{(i)}$ is a string of zeros only. Those $D^{(i)}$s in every decoding step i form a set S_i of decoding candidates.

It is obvious that this step is the most crucial iterative part of the decoder. During the process of decoding, the number of elements in the sets S_i of all possible candidates can increase dramatically, so it is important to keep this number under control. Positioning of the redundant data in

Table 3: The left parastrophe $Qpar(\backslash)$ of the quasigroup $Q(*)$

\	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	8	7	2	0	d	3	6	5	c	e	f	9	1	a	b	4
1	0	5	a	1	f	9	8	6	4	2	b	7	c	3	e	d
2	1	0	f	9	4	5	a	b	d	7	c	e	3	8	2	6
3	b	3	c	8	5	f	0	9	a	4	7	1	d	e	6	2
4	2	f	9	7	0	1	4	3	b	6	a	5	e	c	d	8
5	3	2	5	a	6	c	f	8	e	d	1	b	7	9	4	0
6	7	d	0	3	b	e	c	f	5	a	2	8	4	6	9	1
7	d	4	b	f	c	8	7	e	6	1	3	a	2	5	0	9
8	9	8	3	e	a	7	2	1	f	b	4	6	0	d	c	5
9	f	6	e	5	2	a	b	c	8	3	d	0	9	4	1	7
a	4	9	d	b	1	6	5	7	3	0	e	c	f	2	8	a
b	a	e	4	6	7	2	9	0	1	f	5	d	8	b	3	c
c	6	c	1	d	e	0	3	4	9	5	8	2	a	f	7	b
d	c	a	8	4	3	b	1	d	2	9	0	f	6	7	5	e
e	5	1	6	2	8	d	e	a	7	c	9	4	b	0	f	3
f	e	b	7	c	9	4	d	2	0	8	6	3	5	1	a	f

L, as shown in Table 1, is used for this purpose, but also other techniques for eliminating the most unlikely candidates can be applied. At the end of the iterative decoding, eventually the number of elements in S_i decreases to one, meaning that all errors are found and corrected. The decoder has also the following two properties: (i) (*wrong decision detection*) if somehow the right candidate is eliminated from the set of candidates S_i, several steps further the decoding process will eventually result in an empty set S_j, which is an evidence that some wrong decoding decision was made, and (ii) (*early decoding property*) since the decoding is done iteratively with equation (2), decoding of correct bits starts much earlier: for example, even if the block length is 864 bits, the decision on correct bits will start just after decoding approximately 60 bits. This property gives another unique property of our code: if the decoding is done with some errors, large part of the decoded block has correct information.

Our numerical experiments are summarized on Figure 1. On Figure 1 we present the results of our (216,864) code with rate 1/4 compared with the JPL implementation of Turbo Codes of length 65536 bits and irregular LDPC codes over GF(8) with length of 48000 bits (the data for Turbo and LDPC codes are from [10] p.568).

4. Conclusions

In this work we have designed a class of error correction codes, generated by quasigroup string transformations. Although the software implementation of our code is far from optimized, numerical experiments show that its potentials are, in this moment, far beyond the capabilities of the currently best error correction codes.

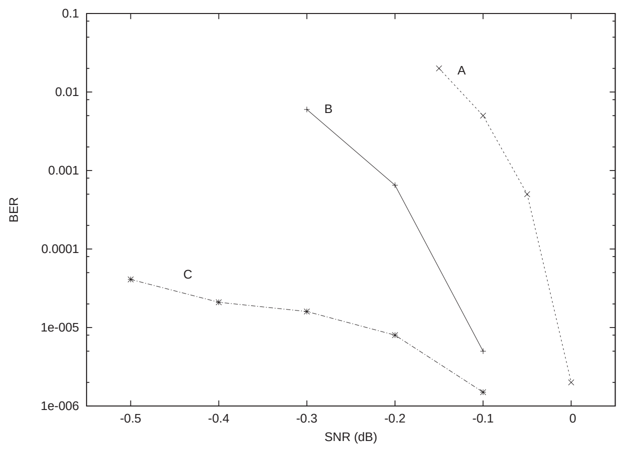

Figure 1: A) JPL Turbo Code, 1/4 rate with block length of 65536 bits, B) LDPC GF(8), 1/4 rate with block length of 48000 bits, C) Numerical results of our 1/4 rate code with block length of 864 bits.

References

[1] C. E. Shannon, "A Mathematical Theory of Communication", *Bell Sys. Tech. J.*, Vol. 26, pp. 379–423, pp. 623–656, 1948.

[2] S. Franz, M. Leone, A. Montanari, and F. Ricci-Tersenghi, "Dynamic phase transition for decoding algorithms," *Physical Review* E vol. 66, pp. 046120, 2002.

[3] N. S. Skantzos, J. van Mourik, and D. Saad, "Magnetization enumerator of real-valued symmetric channels in Gallager error-correcting codes," *Physical Review* E vol. 67, pp. 037101, 2003.

[4] K. Nakamura, Y. Kabashima, R. Morelos-Zaragoza, and D. Saad, "Statistical mechanics of broadcast channels using low-density parity-check codes," *Physical Review* E vol. 67, pp. 036703, 2003.

[5] L. Kocarev, Z. Tasev, and A. Vardy, "Improving turbo codes by control of transient chaos in turbo-decoding algorithms," *Electronics Letters*, Vol. 38(20), pp. 1184–1186, 2002.

[6] Jung-Hoon Kim and Young-Jo Ko, "Error-correcting codes on scale-free networks," *Physical Review* E vol. 69, pp. 067103, 2004.

[7] T. Richardson, "The geometry of turbo decoding dynamics," *IEEE Trans. Inform. Theory*, vol. 46, pp. 9–23, 2000.

[8] J. Dénes, A.D. Keedwell, *Latin Squares and their Applications*, English Univer. Press Ltd., 1974.

[9] S. Markovski, D. Gligoroski, V. Bakeva, "Quasigroup String Processing: Part 1," *Maced. Acad. of Sci. and Arts, Sec. Math. Tech. Scien.* vol. XX 1-2, 1999, pp 13–28, 1999.

[10] D.J.C. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press 2003.